

BIOINFORMATICS

FOURTH EDITION

EDITED BY

ANDREAS D. BAXEVANIS

GARY D. BADER

DAVID S. WISHART



WILEY

Bioinformatics

Bioinformatics

Edited by

Andreas D. Baxevanis, Gary D. Bader, and David S. Wishart

Fourth Edition

WILEY

This fourth edition first published 2020
© 2020 John Wiley & Sons, Inc.

Edition History

Wiley-Blackwell (1e, 2000), Wiley-Blackwell (2e, 2001), Wiley-Blackwell (3e, 2005)

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, except as permitted by law. Advice on how to obtain permission to reuse material from this title is available at <http://www.wiley.com/go/permissions>.

The right of Andreas D. Baxevanis, Gary D. Bader, and David S. Wishart to be identified as the authors of the editorial material in this work has been asserted in accordance with law.

Registered Office

John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, USA

Editorial Office

John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, USA

For details of our global editorial offices, customer services, and more information about Wiley products visit us at www.wiley.com.

Wiley also publishes its books in a variety of electronic formats and by print-on-demand. Some content that appears in standard print versions of this book may not be available in other formats.

Limit of Liability/Disclaimer of Warranty

While the publisher and authors have used their best efforts in preparing this work, they make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives, written sales materials or promotional statements for this work. The fact that an organization, website, or product is referred to in this work as a citation and/or potential source of further information does not mean that the publisher and authors endorse the information or services the organization, website, or product may provide or recommendations it may make. This work is sold with the understanding that the publisher is not engaged in rendering professional services. The advice and strategies contained herein may not be suitable for your situation. You should consult with a specialist where appropriate. Further, readers should be aware that websites listed in this work may have changed or disappeared between when this work was written and when it is read. Neither the publisher nor authors shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

Library of Congress Cataloging-in-Publication Data

Names: Baxevanis, Andreas D., editor. | Bader, Gary D., editor. | Wishart, David S., editor.

Title: Bioinformatics / edited by Andreas D. Baxevanis, Gary D. Bader, David S. Wishart.

Other titles: Bioinformatics (Baxevanis)

Description: Fourth edition. | Hoboken, NJ : Wiley, 2020. | Includes bibliographical references and index.

Identifiers: LCCN 2019030489 (print) | ISBN 9781119335580 (cloth) | ISBN 9781119335962 (adobe pdf) | ISBN 9781119335955 (epub)

Subjects: MESH: Computational Biology--methods | Sequence Analysis--methods | Base Sequence | Databases, Nucleic Acid | Databases, Protein

Classification: LCC QH324.2 (print) | LCC QH324.2 (ebook) | NLM QU 550.5.S4 | DDC 570.285--dc23

LC record available at <https://lcn.loc.gov/2019030489>

LC ebook record available at <https://lcn.loc.gov/2019030490>

Cover Design: Wiley

Cover Images: © David Wishart, background © Suebsiri/Getty Images

Set in 9.5/12.5pt STIXTwoText by SPi Global, Chennai, India

Contents

Foreword *vii*

Preface *ix*

Contributors *xi*

About the Companion Website *xvii*

- 1 Biological Sequence Databases** 1
Andreas D. Baxevanis
- 2 Information Retrieval from Biological Databases** 19
Andreas D. Baxevanis
- 3 Assessing Pairwise Sequence Similarity: BLAST and FASTA** 45
Andreas D. Baxevanis
- 4 Genome Browsers** 79
Tyra G. Wolfsberg
- 5 Genome Annotation** 117
David S. Wishart
- 6 Predictive Methods Using RNA Sequences** 155
Michael F. Sloma, Michael Zuker, and David H. Mathews
- 7 Predictive Methods Using Protein Sequences** 185
Jonas Reeb, Tatyana Goldberg, Yanay Ofra, and Burkhard Rost
- 8 Multiple Sequence Alignments** 227
Fabian Sievers, Geoffrey J. Barton, and Desmond G. Higgins
- 9 Molecular Evolution and Phylogenetic Analysis** 251
Emma J. Griffiths and Fiona S.L. Brinkman
- 10 Expression Analysis** 279
Marieke L. Kuijjer, Joseph N. Paulson, and John Quackenbush
- 11 Proteomics and Protein Identification by Mass Spectrometry** 315
Sadhna Phanse and Andrew Emili
- 12 Protein Structure Prediction and Analysis** 363
David S. Wishart
- 13 Biological Networks and Pathways** 399
Gary D. Bader

14	Metabolomics	437
	<i>David S. Wishart</i>	
15	Population Genetics	481
	<i>Lynn B. Jorde and W. Scott Watkins</i>	
16	Metagenomics and Microbial Community Analysis	505
	<i>Robert G. Beiko</i>	
17	Translational Bioinformatics	537
	<i>Sean D. Mooney and Stephen J. Mooney</i>	
18	Statistical Methods for Biologists	555
	<i>Hunter N.B. Moseley</i>	
	Appendices	583
	Glossary	591
	Index	609

Foreword

As I review the material presented in the fourth edition of *Bioinformatics* I am moved in two ways, related to both the past and the future.

Looking to the past, I am moved by the amazing evolution that has occurred in our field since the first edition of this book appeared in 1998. Twenty-one years is a long, long time in any scientific field, but especially so in the agile field of bioinformatics. To use the well-trodden metaphor of the “biology moonshot,” the launchpad at the beginning of the twenty-first century was the determination of the human genome. Discovery is not the right word for what transpired – we knew it was there and what was needed. Synergy is perhaps a better word; synergy of technological development, experiment, computation, and policy. A truly collaborative effort to continuously share, in a reusable way, the collective efforts of many scientists. Bioinformatics was born from this synergy and has continued to grow and flourish based on these principles.

That growth is reflected in both the scope and depth of what is covered in these pages. These attributes are a reflection of the increased complexity of the biological systems that we study (moving from “simple” model organisms to the human condition) and the scales at which those studies take place. As a community we have professed multiscale modeling without much to show for it, but it would seem to be finally here. We now have the ability to connect the dots from molecular interactions, through the pathways to which those molecules belong to the cells they affect, to the interactions between those cells through to the effects they have on individuals within a population. Tools and methodologies that were novel in earlier editions of this book are now routine or obsolete, and newer, faster, and more accurate procedures are now with us. This will continue, and as such this book provides a valuable snapshot of the scope and depth of the field as it exists today.

Looking to the future, this book provides a foundation for what is to come. For me this is a field more aptly referred to (and perhaps a new subtitle for the next edition) as Biomedical Data Science. Sitting as I do now, as Dean of a School of Data Science which collaborates openly across all disciplines, I see rapid change akin to what happened to birth bioinformatics 20 or more years ago. It will not take 20 years for other disciplines to catch up; I predict it will take 2! The accomplishments outlined in this book can help define what other disciplines will accomplish with their own data in the years to come. Statistical methods, cloud computing, data analytics, notably deep learning, the management of large data, visualization, ethics policy, and the law surrounding data are generic. Bioinformatics has so much to offer, yet it will also be influenced by other fields in a way that has not happened before. Forty-five years in academia tells me that there is nothing to compare across campuses to what is happening today. This is both an opportunity and a threat. The editors and authors of this edition should be complimented for setting the stage for what is to come.

Philip E. Bourne, University of Virginia

Preface

In putting together this textbook, we hope that students from a range of fields – including biology, computer science, engineering, physics, mathematics, and statistics – benefit by having a convenient starting point for learning most of the core concepts and many useful practical skills in the field of bioinformatics, also known as computational biology.

Students interested in bioinformatics often ask about how should they acquire training in such an interdisciplinary field as this one. In an ideal world, students would become experts in all the fields mentioned above, but this is actually not necessary and realistically too much to ask. All that is required is to combine their scientific interests with a foundation in biology and any single quantitative field of their choosing. While the most common combination is to mix biology with computer science, incredible discoveries have been made through finding creative intersections with any number of quantitative fields. Indeed, many of these quantitative fields typically overlap a great deal, especially given their foundational use of mathematics and computer programming. These natural relationships between fields provide the foundation for integrating diverse expertise and insights, especially when in the context of performing bioinformatic analyses.

While bioinformatics is often considered an independent subfield of biology, it is likely that the next generation of biologists will not consider bioinformatics as being separate and will instead consider gaining bioinformatics and data science skills as naturally as they learn how to use a pipette. They will learn how to program a computer, likely starting in elementary school. Other data science knowledge areas, such as math, statistics, machine learning, data processing, and data visualization will also be part of any core curriculum. Indeed, the children of one of the editors recently learned how to construct bar plots and other data charts in kindergarten! The same editor is teaching programming in R (an important data science programming language) to all incoming biology graduate students at his university starting this year.

As bioinformatics and data science become more naturally integrated in biology, it is worth noting that these fields actively espouse a culture of open science. This culture is motivated by thinking about why we do science in the first place. We may be curious or like problem solving. We could also be motivated by the benefits to humanity that scientific advances bring, such as tangible health and economic benefits. Whatever the motivating factor, it is clear that the most efficient way to solve hard problems is to work together as a team, in a complementary fashion and without duplication of effort. The only way to make sure this works effectively is to efficiently share knowledge and coordinate work across disciplines and research groups. Presenting scientific results in a reproducible way, such as freely sharing the code and data underlying the results, is also critical. Fortunately, there are an increasing number of resources that can help facilitate these goals, including the bioRxiv preprint server, where papers can be shared before the very long process of peer review is completed; GitHub, for sharing computer code; and data science notebook technology that helps combine code, figures, and text in a way that makes it easier to share reproducible and reusable results.

We hope this textbook helps catalyze this transition of biology to a quantitative, data science-intensive field. As biological research advances become ever more built on interdisciplinary, open, and team science, progress will dramatically speed up, laying the groundwork for fantastic new discoveries in the future.

We also deeply thank all of the chapter authors for contributing their knowledge and time to help the many future readers of this book learn how to apply the myriad bioinformatic techniques covered within these pages to their own research questions.

Andreas D. Baxevanis

Gary D. Bader

David S. Wishart

Contributors

Gary D. Bader, PhD is a Professor at The Donnelly Centre at the University of Toronto, Toronto, Canada, and a leader in the field of Network Biology. Gary completed his postdoctoral work in Chris Sander's group in the Computational Biology Center (cBio) at Memorial Sloan-Kettering Cancer Center in New York. Gary completed his PhD in the laboratory of Christopher Hogue in the Department of Biochemistry at the University of Toronto and a BSc in Biochemistry at McGill University in Montreal. Dr. Bader uses molecular interaction, pathway, and -omics data to gain a "causal" mechanistic understanding of normal and disease phenotypes. His laboratory develops novel computational approaches that combine molecular interaction and pathway information with -omics data to develop clinically predictive models and identify therapeutically targetable pathways. He also helps lead the Cytoscape, GeneMANIA, and Pathway Commons pathway and network analysis projects.

Geoffrey J. Barton, PhD is Professor of Bioinformatics and Head of the Division of Computational Biology at the University of Dundee School of Life Sciences, Dundee, UK. Before moving to Dundee in 2001, he was Head of the Protein Data Bank in Europe and the leader of the Research and Development Team at the EMBL European Bioinformatics Institute (EBI). Prior to joining EMBL-EBI, he was Head of Genome Informatics at the Wellcome Trust Centre for Human Genetics, University of Oxford, a position he held concurrently with a Royal Society University Research Fellowship in the Department of Biochemistry. Geoff's longest running research interest is using computational methods to study the relationship between a protein's sequence, its structure, and its function. His group has contributed many tools and techniques in the field of protein sequence and structure analysis and structure prediction. Two of the best known are the Jalview multiple alignment visualization and analysis workbench, which is in use by over 70 000 groups for research and teaching, and the JPred multi-neural net protein secondary structure prediction algorithm, which performs predictions on up to 500 000 proteins/month for users worldwide. In addition to his work related to protein sequence and structure, Geoff has collaborated on many projects that probe biological processes using proteomic and high-throughput sequencing approaches. Geoff's group has deep expertise in RNA-seq methods and has recently published a two-condition 48-replicate RNA-seq study that is now a key reference work for users of this technology.

Andreas D. Baxevanis, PhD is the Director of Computational Biology for the National Institutes of Health's (NIH) Intramural Research Program. He is also a Senior Scientist leading the Computational Genomics Unit at the NIH's National Human Genome Research Institute, Bethesda, MD, USA. His research program is centered on probing the interface between genomics and developmental biology, focusing on the sequencing and analysis of invertebrate genomes that can yield insights of relevance to human health, particularly in the areas of regeneration, allorecognition, and stem cell biology. His accomplishments have been recognized by the Bodossaki Foundation's Academic Prize in Medicine and Biology in 2000,

Greece's highest award for young scientists of Greek heritage. In 2014, he was elected to the Johns Hopkins Society of Scholars, recognizing alumni who have achieved marked distinction in their field of study. He was the recipient of the NIH's Ruth L. Kirschstein Mentoring Award in 2015, in recognition of his commitment to scientific training, education, and mentoring. In 2016, Dr. Baxevanis was elected as a Senior Member of the International Society for Computational Biology for his sustained contributions to the field and, in 2018, he was elected as a Fellow of the American Association for the Advancement of Science for his distinguished contributions to the field of comparative genomics.

Robert G. Beiko, PhD is a Professor and Associate Dean for Research in the Faculty of Computer Science at Dalhousie University, Halifax, Nova Scotia, Canada. He is a former Tier II Canada Research Chair in Bioinformatics (2007–2017), an Associate Editor at *mSystems* and *BMC Bioinformatics*, and a founding organizer of the Canadian Bioinformatics Workshops in Metagenomics and Genomic Epidemiology. He is also the lead editor of the recently published book *Microbiome Analysis* in the *Methods in Molecular Biology* series. His research focuses on microbial genomics, evolution, and ecology, with concentrations in the area of lateral gene transfer and microbial community analysis.

Fiona S.L. Brinkman, PhD, FRSC is a Professor in Bioinformatics and Genomics in the Department of Molecular Biology and Biochemistry at Simon Fraser University, Vancouver, British Columbia, Canada, with cross-appointments in Computing Science and the Faculty of Health Sciences. She is most known for her research and development of widely used computer software that aids both microbe (PSORTb, IslandViewer) and human genomic (InnateDB) evolutionary/genomics analyses, along with her insights into pathogen evolution. She is currently co-leading a national effort – the Integrated Rapid Infectious Disease Analysis Project – the goal of which is to use microbial genomes as a fingerprint to better track and understand the spread and evolution of infectious diseases. She has also been leading development into an approach to integrate very diverse data for the Canadian CHILD Study birth cohort, including microbiome, genomic, epigenetic, environmental, and social data. She coordinates community-based genome annotation and database development for resources such as the *Pseudomonas* Genome Database. She also has a strong interest in bioinformatics education, including developing the first undergraduate curricula used as the basis for the first White Paper on Canadian Bioinformatics Training in 2002. She is on several committees and advisory boards, including the Board of Directors for Genome Canada; she chairs the Scientific Advisory Board for the European Nucleotide Archive (EMBL-EBI). She has received a number of awards, including a TR100 award from MIT, and, most recently, was named as a Fellow of the Royal Society of Canada.

Andrew Emili, PhD is a Professor in the Departments of Biochemistry (Medical School) and Biology (Arts and Sciences) at Boston University (BU), Boston, MA, USA, and the inaugural Director of the BU Center for Network Systems Biology (CNSB). Prior to Boston, Dr. Emili was a founding member and Principal Investigator for 18 years at the Donnelly Center for Cellular and Biomolecular Research at the University of Toronto, one of the premier research centers in integrative molecular biology. Dr. Emili is an internationally recognized leader in functional proteomics, systems biology, and precision mass spectrometry. His group develops and applies innovative technologies to systematically map protein interaction networks and macromolecular complexes of cells and tissues on a global scale, publishing “interactome” maps of unprecedented quality, scope, and resolution.

Tatyana Goldberg, PhD is a postdoctoral scientist at the Technical University of Munich, Germany. She obtained her PhD in Bioinformatics under the supervision of Dr. Burkhard Rost. Her research focuses on developing models that can predict the localization of proteins within cells. The results of her study contribute to a variety of applications, including the development of pharmaceuticals for the treatment of Alzheimer disease and cancer.

Emma J. Griffiths, PhD is a research associate in the Department of Pathology and Laboratory Medicine at the University of British Columbia in Vancouver, Canada, working with Dr. William Hsiao. Dr. Griffiths received her PhD from the Department of Biochemistry and Biomedical Sciences at McMaster University in Hamilton, Canada, with her doctoral work focusing on the evolutionary relationships between different groups of bacteria. She has since pursued postdoctoral training in the fields of chemical and fungal genetics and microbial genomics with Dr. Fiona Brinkman in the Department of Biochemistry and Molecular Biology at Simon Fraser University in Vancouver, Canada. Her current work focuses on the development of ontology-driven applications designed to improve pathogen genomics contextual data (“metadata”) exchange during public health investigations.

Desmond G. Higgins, PhD is Professor of Bioinformatics in University College Dublin, Ireland, where his laboratory works on genomic data analysis and sequence alignment algorithms. He earned his doctoral degree in zoology from Trinity College Dublin, Ireland, and has worked in the field of bioinformatics since 1985. His group maintains and develops the Clustal package for multiple sequence alignment in collaboration with groups in France, Germany, and the United Kingdom. Dr. Higgins wrote the first version of Clustal in Dublin in 1988. He then moved to the EMBL Data Library group located in Heidelberg in 1990 and later to EMBL-EBI in Hinxton. This coincided with the release of ClustalW and, later, ClustalX, which has been extremely widely used and cited. Currently, he has run out of version letters so is working on Clustal Omega, specifically designed for making extremely large protein alignments.

Lynn B. Jorde, PhD has been on the faculty of the University of Utah School of Medicine, Salt Lake City, UT, USA, since 1979 and holds the Mark and Kathie Miller Presidential Endowed Chair in Human Genetics. He was appointed Chair of the Department of Human Genetics in September 2009. Dr. Jorde’s laboratory has published scientific articles on human genetic variation, high-altitude adaptation, the genetic basis of human limb malformations, and the genetics of common diseases such as hypertension, juvenile idiopathic arthritis, and inflammatory bowel disease. Dr. Jorde is the lead author of *Medical Genetics*, a textbook that is now in its fifth edition and translated into multiple foreign languages. He is the co-recipient of the 2008 Award for Excellence in Education from the American Society of Human Genetics (ASHG). He served two 3-year terms on the Board of Directors of ASHG and, in 2011, he was elected as president of ASHG. In 2012, he was elected as a Fellow of the American Association for the Advancement of Science.

Marieke L. Kuijjer, PhD is a Group Leader at the Centre for Molecular Medicine Norway (NCMM, a Nordic EMBL partner), University of Oslo, Norway, where she runs the Computational Biology and Systems Medicine group. She obtained her doctorate in the laboratory of Dr. Pancras Hogendoorn in the Department of Pathology at the Leiden University Medical Center in the Netherlands. After this, she continued her scientific training as a postdoctoral researcher in the laboratory of Dr. John Quackenbush at the Dana-Farber Cancer Institute and Harvard T.H. Chan School of Public Health, during which she won a career development award and a postdoctoral fellowship. Dr. Kuijjer’s research focuses on solving fundamental biological questions through the development of new methods in computational and systems biology and on implementing these techniques to better understand gene regulation in cancer. Dr. Kuijjer serves on the editorial board of *Cancer Research*.

David H. Mathews, MD, PhD is a professor of Biochemistry and Biophysics and also of Biostatistics and Computational Biology at the University of Rochester Medical Center, Rochester, NY, USA. He also serves as the Associate Director of the University of Rochester’s Center for RNA Biology. His involvement in education includes directing the Biophysics PhD program and teaching a course in Python programming and algorithms for doctoral students without a programming background. His group studies RNA biology and develops methods

for RNA secondary structure prediction and molecular modeling of three-dimensional structure. His group developed and maintains RNAstructure, a widely used software package for RNA structure prediction and analysis.

Sean D. Mooney, PhD has spent his career as a researcher and group leader in biomedical informatics. He now leads Research IT for UW Medicine and is leading efforts to support and build clinical research informatic platforms as its first Chief Research Information Officer (CRIO) and as a Professor in the Department of Biomedical Informatics and Medical Education at the University of Washington, Seattle, WA, USA. Previous to being appointed as CRIO, he was an Associate Professor and Director of Bioinformatics at the Buck Institute for Research on Aging. As an Assistant Professor, he was appointed in Medical and Molecular Genetics at Indiana University School of Medicine and was the founding Director of the Indiana University School of Medicine Bioinformatics Core. In 1997, he received his BS with Distinction in Biochemistry and Molecular Biology from the University of Wisconsin at Madison. He received his PhD from the University of California in San Francisco in 2001, then pursued his postdoctoral studies under an American Cancer Society John Peter Hoffman Fellowship at Stanford University.

Stephen J. Mooney, PhD is an Acting Assistant Professor in the Department of Epidemiology at the University of Washington, Seattle, WA, USA. He developed the CANVAS system for collecting data from Google Street View imagery as a graduate student, and his research focuses on contextual influences on physical activity and transport-related injury. He's a methods geek at heart.

Hunter N.B. Moseley, PhD is an Associate Professor in the Department of Molecular and Cellular Biochemistry at the University of Kentucky, Lexington, KY, USA. He is also the Informatics Core Director within the Resource Center for Stable Isotope Resolved Metabolomics, Associate Director for the Institute for Biomedical Informatics, and a member of the Markey Cancer Center. His research interests include developing computational methods, tools, and models for analyzing and interpreting many types of biological and biophysical data that enable new understanding of biological systems and related disease processes. His formal education spans multiple disciplines including chemistry, mathematics, computer science, and biochemistry, with expertise in algorithm development, mathematical modeling, structural bioinformatics, and systems biochemistry, particularly in the development of automated analyses of nuclear magnetic resonance and mass spectrometry data as well as knowledge–data integration.

Yanay Ofran, PhD is a Professor and head of the Laboratory of Functional Genomics and Systems Biology at Bar Ilan University in Tel Aviv, Israel. His research focuses on biomolecular recognition and its role in health and disease. Professor Ofran is also the founder of Biologic Design, a biopharmaceutical company that uses artificial intelligence approaches to design epitope-specific antibodies. He is also the co-founder of Ukko, a biotechnology company that uses computational tools to design safe proteins for the food and agriculture sectors.

Joseph N. Paulson, PhD is a Statistical Scientist within Genentech's Department of Biostatistics, San Francisco, CA, USA, working on designing clinical trials and biomarker discovery. Previously, he was a Research Fellow in the Department of Biostatistics and Computational Biology at the Dana-Farber Cancer Institute and Department of Biostatistics at the Harvard T.H. Chan School of Public Health. He graduated with a PhD in Applied Mathematics, Statistics, and Scientific Computation from the University of Maryland, College Park where he was a National Science Foundation Graduate Fellow. As a statistician and computational biologist, his interests include clinical trial design, biomarker discovery,

development of computational methods for the analysis of high-throughput sequencing data while accounting for technical artifacts, and the microbiome.

Sadhna Phanse, MSc is a Bioinformatics Analyst at the Donnelly Centre for Cellular and Biomolecular Research at the University of Toronto, Toronto, Canada. She has been active in the field of proteomics since 2006 as a member of the Emili research group. Her current work involves the use of bioinformatics methods to investigate biological systems and molecular association networks in human cells and model organisms.

John Quackenbush, PhD is Professor of Computational Biology and Bioinformatics and Chair of the Department of Biostatistics at the Harvard T.H. Chan School of Public Health, Boston, MA, USA. He also holds appointments in the Channing Division of Network Medicine of Brigham and Women's Hospital and at the Dana-Farber Cancer Institute. He is a recognized expert in computational and systems biology and its applications to the study of a wide range of human diseases and the factors that drive those diseases and their responses to therapy. Dr. Quackenbush has long been an advocate for open science and reproducible research. As a founding member and past president of the Functional Genomics Data Society (FGED), he was a developer of the Minimal Information About a Microarray Experiment (MIAME) and other data-reporting standards. Dr. Quackenbush was honored by President Barack Obama in 2013 as a White House Open Science Champion of Change.

Jonas Reeb, MSc is a PhD student in the laboratory of Burkhard Rost at the Technical University of Munich, Germany (TUM). During his studies at TUM, he has worked on predictive methods for the analysis and evaluation of transmembrane proteins; he has also worked on the NYCOMPS structural genomics pipeline. His doctoral thesis focuses on the effect of sequence variants and their prediction.

Burkhard Rost, PhD is a professor and Alexander von Humboldt Award recipient at the Technical University of Munich, Germany (TUM). He was the first to combine machine learning with evolutionary information, using this combination to accurately predict secondary structure. Since that time, his group has repeated this success in developing many other tools that are actively used to predict and understand aspects of protein structure and function. All tools developed by his research group are available through the first internet server in the field of protein structure prediction (PredictProtein), a resource that has been online for over 25 years. Over the last several years, his research group has been shifting its focus to the development of methods that predict and annotate the effect of sequence variation and their implications for precision medicine and personalized health.

Fabian Sievers, PhD is currently a postdoctoral research fellow in the laboratory of Des Higgins at University College Dublin, Ireland. He works on multiple sequence alignment algorithms and, in particular, on the development of Clustal Omega. He received his PhD in mathematics from Trinity College, Dublin and has worked in industry in the fields of algorithm development and high-performance computing.

Michael F. Sloma, PhD is a data scientist at Xometry, Gaithersburg, MD, USA. He received his BA degree in Chemistry from Wells College. He earned his doctoral degree in Biochemistry in the laboratory of David Mathews at the University of Rochester, where his research focused on computational methods to predict RNA structure from sequence.

W. Scott Watkins, MS is a researcher and laboratory manager in the Department of Human Genetics at the University of Utah, Salt Lake City, UT, USA. He has a long-standing interest in human population genetics and evolution. His current interests include the development and application of high-throughput computational methods to mobile element biology, congenital heart disease, and personalized medicine.

David S. Wishart, PhD is a Distinguished University Professor in the Departments of Biological Sciences and Computing Science at the University of Alberta, Edmonton, Alberta, Canada. Dr. Wishart has been developing bioinformatics programs and databases since the early 1980s and has made bioinformatics an integral part of his research program for nearly four decades. His interest in bioinformatics led to the development of a number of widely used bioinformatics tools for structural biology, bacterial genomics, pharmaceutical research, and metabolomics. Some of Dr. Wishart's most widely known bioinformatics contributions include the Chemical Shift Index (CSI) for protein secondary structure identification by nuclear magnetic resonance spectroscopy, PHAST for bacterial genome annotation, the DrugBank database for drug research, and MetaboAnalyst for metabolomic data analysis. Over the course of his academic career, Dr. Wishart has published more than 400 research papers, with many being in the field of bioinformatics. In addition to his long-standing interest in bioinformatics research, Dr. Wishart has been a passionate advocate for bioinformatics education and outreach. He is one of the founding members of the Canadian Bioinformatics Workshops (CBW) – a national bioinformatics training program that has taught more than 3000 students over the past two decades. In 2002 he established Canada's first undergraduate bioinformatics degree program at the University of Alberta and has personally mentored nearly 130 undergraduate and graduate students, many of whom have gone on to establish successful careers in bioinformatics.

Tyra G. Wolfsberg, PhD is the Associate Director of the Bioinformatics and Scientific Programming Core at the National Human Genome Research Institute (NHGRI), National Institutes of Health (NIH), Bethesda, MD, USA. Her research program focuses on developing methodologies to integrate sequence, annotation, and experimentally generated data so that bench biologists can quickly and easily obtain results for their large-scale experiments. She maintains a long-standing commitment to bioinformatics education and outreach. She has authored a chapter on genomic databases for previous editions of this textbook, as well as a chapter on the NCBI MapViewer for *Current Protocols in Bioinformatics* and *Current Protocols in Human Genetics*. She serves as the co-chair of the NIH lecture series Current Topics in Genome Analysis; these lectures are archived online and have been viewed over 1 million times to date. In addition to teaching bioinformatics courses at NHGRI, she served for 13 years as a faculty member in bioinformatics at the annual AACR Workshop on Molecular Biology in Clinical Oncology.

Michael Zuker, PhD retired as a Professor of Mathematical Sciences at Rensselaer Polytechnic Institute, Troy, NY, USA, in 2016. He was an Adjunct Professor in the RNA Institute at the University of Albany and remains affiliated with the RNA Institute. He works on the development of algorithms to predict folding, hybridization, and melting profiles in nucleic acids. His nucleic acid folding and hybridization web servers have been running at the University of Albany since 2010. His educational activities include developing and teaching his own bioinformatics course at Rensselaer and participating in both a Chautauqua short course in bioinformatics for college teachers and an intensive bioinformatics course at the University of Michigan. He currently serves on the Scientific Advisory Board of Expansion Therapeutics, Inc. at the Scripps Research Institute in Jupiter, Florida.

About the Companion Website

This book is accompanied by a companion website:

www.wiley.com/go/baxevanis/Bioinformatics_4e



The website includes:

- Test Samples
- Word Samples

Scan this QR code to visit the companion website.



1

Biological Sequence Databases

Andreas D. Baxevanis

Introduction

Over the past several decades, there has been a feverish push to understand, at the most elementary of levels, what constitutes the basic “book of life.” Biologists (and scientists in general) are driven to understand how the millions or billions of bases in an organism’s genome contain all of the information needed for the cell to conduct the myriad metabolic processes necessary for the organism’s survival – information that is propagated from generation to generation. To have a basic understanding of how the collection of individual nucleotide bases drives the engine of life, large amounts of sequence data must be collected and stored in a way that these data can be searched and analyzed easily. To this end, much effort has gone into the design and maintenance of biological sequence databases. These databases have had a significant impact on the advancement of our understanding of biology not just from a computational standpoint but also through their integrated use alongside studies being performed at the bench.

The history of sequence databases began in the early 1960s, when Margaret Dayhoff and colleagues (1965) at the National Biomedical Research Foundation (NBRF) collected all of the protein sequences known at that time – all 65 of them – and published them in a book called the *Atlas of Protein Sequence and Structure*. It is important to remember that, at this point in the history of biology, the focus was on sequencing proteins through traditional techniques such as the Edman degradation rather than on sequencing DNA, hence the overall small number of available sequences. By the late 1970s, when a significant number of nucleotide sequences became available, those were also included in later editions of the *Atlas*. As this collection evolved, it included text-based descriptions to accompany the protein sequences, as well as information regarding the evolution of many protein families. This work, in essence, was the first annotated sequence database, even though it was in printed form. Over time, the amount of data contained in the *Atlas* became unwieldy and the need for it to be available in electronic form became obvious. From the early 1970s to the late 1980s, the contents of the *Atlas* were distributed electronically by NBRF (and later by the Protein Information Resource, or PIR) on magnetic tape, and the distribution included some basic programs that could be used to search and evaluate distant evolutionary relationships.

The next phase in the history of sequence databases was precipitated by the veritable explosion in the amount of nucleotide sequence data available to researchers by the end of the 1970s. To address the need for more robust public sequence databases, the Los Alamos National Laboratory (LANL) created the Los Alamos DNA Sequence Database in 1979, which became known as GenBank in 1982 (Benson et al. 2018). Meanwhile, the European Molecular Biology Laboratory (EMBL) created the EMBL Nucleotide Sequence Data Library in 1980. Throughout the 1980s, EMBL (then based in Heidelberg, Germany), LANL, and (later) the National Center for Biotechnology Information (NCBI, part of the National Library of Medicine at the National Institutes of Health) jointly contributed DNA sequence data to these databases. This was done

by having teams of curators manually transcribing and interpreting what was published in print journals to an electronic format more appropriate for computational analyses. The DNA Databank of Japan (DDBJ; Kodama et al. 2018) joined this DNA data-collecting collaboration a few years later. By the late 1980s, the quantity of DNA sequence data being produced was so overwhelming that print journals began asking scientists to electronically submit their DNA sequences directly to these databases, rather than publishing them in printed journals or papers. In 1988, after a meeting of these three groups (now referred to as the International Nucleotide Sequence Database Collaboration, or INSDC; Karsch-Mizrachi et al. 2018), there was an agreement to use a common data exchange format and to have each database update only the records that were directly submitted to it. Thanks to this agreement, all three centers (EMBL, DDBJ, and NCBI) now collect direct DNA sequence submissions and distribute them so that each center has copies of all of the sequences, with each center acting as a primary distribution center for these sequences. DDBJ/EMBL/GenBank records are updated automatically every 24 hours at all three sites, meaning that all sequences can be found within DDBJ, the European Nucleotide Archive (ENA; Silvester et al. 2018), and GenBank in short order. That said, each database within the INSDC has the freedom to display and annotate the sequence data as it sees fit.

In parallel with the early work being done on DNA sequence databases, the foundations for the Swiss-Prot protein sequence database were also being laid in the early 1980s by Amos Bairoch, recounting its history from an engaging perspective in a first-person review (Bairoch 2000). Bairoch converted PIR's *Atlas* to a format similar to that used by EMBL for its nucleotide database. In this initial release, called PIR+, additional information about each of the proteins was added, increasing its value as a curated, well-annotated source of information on proteins. In the summer of 1986, Bairoch began distributing PIR+ on the US BIONET (a precursor to the Internet), renaming it Swiss-Prot. At that time, it contained the grand sum of 3900 protein sequences. This was seen as an overwhelming amount of data, in stark contrast to today's standards. As Swiss-Prot and EMBL followed similar formats, a natural collaboration developed between these two groups, and these collaborative efforts strengthened when both EMBL's and Swiss-Prot's operations were moved to EMBL's European Bioinformatics Institute (EBI; Cook et al. 2018) in Hinxton, UK. One of the first collaborative projects undertaken by the Swiss-Prot and EMBL teams was to create a new and much larger protein sequence database supplement to Swiss-Prot. As maintaining the high quality of Swiss-Prot entries was a time-consuming process involving extensive sequence analysis and detailed curation by expert annotators (Apweiler 2001), and to allow the quick release of protein data not yet annotated to Swiss-Prot's stringent standards, a new database called TrEMBL (for "translation of EMBL nucleotide sequences") was created. This supplement to Swiss-Prot initially consisted of computationally annotated sequence entries derived from the translation of all coding sequences (CDSs) found in INSDC databases. In 2002, a new effort involving the Swiss Institute of Bioinformatics, EMBL-EBI, and PIR was launched, called the UniProt consortium (UniProt Consortium 2017). This effort gave rise to the UniProt Knowledgebase (UniProtKB), consisting of Swiss-Prot, TrEMBL, and PIR. A similar effort also gave rise to the NCBI Protein Database, bringing together data from numerous sources and described more fully in the text that follows.

The completion of human genome sequencing and the sequencing of numerous model genomes, as well as the existence of a gargantuan number of sequences in general, provides a golden opportunity for biological scientists, owing to the inherent value of these data. At the same time, the sheer magnitude of data also presents a conundrum to the inexperienced user, resulting not just from the size of the "sequence information space" but from the fact that the information space continues to get larger by leaps and bounds. Indeed, the sequencing landscape has changed significantly in recent years with the development of new high-throughput technologies that generate more and more sequence data in a way that is best described as "better, cheaper, faster," with these advances feeding into the "insatiable appetite" that scientists have for more and more sequence data (Green et al. 2017). Given the inherent value of the data contained within these sequence databases, this chapter will focus

on providing the reader with a solid understanding of these major public sequence databases, as a first step toward being able to perform robust and accurate bioinformatic analyses.

Nucleotide Sequence Databases

As described above, the major sources of nucleotide sequence data are the databases involved in INSDC – DDBJ, ENA, and GenBank – with new or updated data being shared between these three entities once every 24 hours. This transfer is facilitated by the use of common data formats for the kinds of information described in detail below.

The elementary format underlying the information held in sequence databases is a text file called the *flatfile*. The correspondence between individual flatfile formats greatly facilitates the daily exchange of data between each of these databases. In most cases, fields can be mapped on a one-to-one basis from one flatfile format to the other. Over time, various file formats have been adopted and have found continued widespread use; others have fallen to the wayside for a variety of reasons. The success of a given format depends on its usefulness in a variety of contexts, as well as its power in effectively containing and representing the types of biological data that need to be archived and communicated to scientists.

In its simplest form, a sequence record can be represented as a string of nucleotides with some basic tag or identifier. The most widely used of these simple formats is FASTA, originally introduced as part of the FASTA software suite developed by Lipman and Pearson (1985) that is described in detail in Chapter 3. This inherently simple format provides an easy way of handling primary data for both humans and computers, taking the following form.

```
>U54469.1
CGGTTGCTTGGGTTTATAACATCAGTCAGTGACAGGCATTTCCAGAGTTGCCCTGTTCAACAATCGATA
GCTGCCTTTGGCCACCAAAATCCCAAACCTTAATTAAGAATTAATAATTGGAATAATAATTAAGCCCAG
TAACCTACGCAGCTTGAGTGCCTAACCGATATCTAGTATACATTTGATACATCGAAATCATGGTAGTGT
TGGAGACGGAGAAGGTAAGACGATGATAGACGGCGAGCCGCATGGGTTGATTTGCGCTGAGCCGTGGCA
GGGAACAACAAAAACAGGGTTGTTGCACAAGAGGGGAGGCGATAGTCGAGCGAAAAGAGTGCAGTTGGC
```

For brevity, only the first few lines of the sequence are shown. In the simplest incarnation of the FASTA format, the “greater than” character (>) designates the beginning of a new sequence record; this line is referred to as the *definition line* (commonly called the “def line”). A unique identifier – in this case, the *accession.version number* (U54469.1) – is followed by the nucleotide sequence, in either uppercase or lowercase letters, usually with 60 characters per line. The accession number is the number that is always associated with this sequence (and should be cited in publications), while the version number suffix allows users to easily determine whether they are looking at the most up-to-date record for a particular sequence. The version number suffix is incremented by one each time the sequence is updated.

Additional information can be included on the definition line to make this simple format a bit more informative, as follows.

```
>ENA|U54469|U54469.1 Drosophila melanogaster eukaryotic initiation factor 4E (eIF4E)
gene, complete cds, alternatively spliced.
```

This modified FASTA definition line now has information on the source database (ENA), its accession.version number (U54469.1), and a short description of what biological entity is represented by the sequence.

Nucleotide Sequence Flatfiles: A Dissection

As flatfiles represent the elementary unit of information within sequence databases and facilitate the interchange of information between these databases, it is important to understand

what each individual field within the flatfile represents and what kinds of information can be found in varying parts of the record. While there are minor differences in flatfile formats, they can all be separated into three major parts: the *header*, containing information and descriptors pertaining to the entire record; the *feature table*, which provides relevant annotations to the sequence; and the sequence itself.

The Header

The header is the most database-specific part of the record. Here, we will use the ENA version of the record for discussion (shown in its entirety in Appendix 1.1), with the corresponding DDBJ and GenBank versions of the header appearing in Appendix 1.2. The first line of the record provides basic identifying information about the sequence contained in the record, appropriately named the ID line; this corresponds to the LOCUS line in DDBJ/GenBank.

```
ID   U54469; SV 1; linear; genomic DNA; STD; INV; 2881 BP.
```

The accession number is shown on the ID line, followed by its sequence version (here, the first version, or SV 1). As this is SV 1, this is equivalent to writing U54469.1, as described above. This is then followed by the topology of the DNA molecule (linear) and the molecule type (genomic DNA). The next element represents the ENA data class for this sequence (STD, denoting a “standard” annotated and assembled sequence). Data classes are used to group sequence records within functional divisions, enabling users to query specific subsets of the database. A description of these functional divisions can be found in Box 1.1. Finally, the ID line presents the taxonomic division for the sequence of interest (INV, for invertebrate; see Internet Resources) and its length (2881 base pairs). The accession number will also be shown separately on the AC line that immediately follows the ID lines.

Box 1.1 Functional Divisions in Nucleotide Databases

The organization of nucleotide sequence records into discrete functional types provides a way for users to query specific subsets of the records within these databases. In addition, knowledge that a particular sequence is from a given technique-oriented database allows users to interpret the data from the proper biological point of view. Several of these divisions are described below, and examples of each of these functional divisions (called “data classes” by ENA) can be found by following the example links listed on the ENA Data Formats page listed in the Internet Resources section of this chapter.

- | | |
|-----|---|
| CON | Constructed (or “contigged”) records of chromosomes, genomes, and other long DNA sequences resulting from whole -genome sequencing efforts. The records in this division do not contain sequence data; rather, they contain instructions for the assembly of sequence data found within multiple database records. |
| EST | Expressed Sequence Tags. These records contain short (300–500 bp) single reads from mRNA (cDNA) that are usually produced in large numbers. ESTs represent a snapshot of what is expressed in a given tissue or at a given developmental stage. They represent tags – some coding, some not – of expression for a given cDNA library. |
| GSS | Genome Survey Sequences. Similar to the EST division, except that the sequences are genomic in origin. The GSS division contains (but is not limited to) single-pass read genome survey sequences, bacterial artificial chromosome (BAC) or yeast artificial chromosome (YAC) ends, exon-trapped genomic sequences, and Alu polymerase chain reaction (PCR) sequences. |
| HTG | High-Throughput Genome sequences. Unfinished DNA sequences generated by high-throughput sequencing centers, made available in an expedited fashion to the scientific community for homology and similarity searches. Entries in this division contain keywords indicating its phase within the sequencing process. Once finished, HTG sequences are moved into the appropriate database taxonomic division. |

STD	A record containing a standard, annotated, and assembled sequence.
STS	Sequence-Tagged Sites. Short (200–500 bp) operationally unique sequences that identify a combination of primer pairs used in a PCR assay, generating a reagent that maps to a single position within the genome. The STS division is intended to facilitate cross-comparison of STSs with sequences in other divisions for the purpose of correlating map positions of anonymous sequences with known genes.
WGS	Whole-Genome Shotgun sequences. Sequence data from projects using shotgun approaches that generate large numbers of short sequence reads that can then be assembled by computer algorithms into sequence contigs, higher -order scaffolds, and sometimes into near-chromosome- or chromosome-length sequences.

Following the ID line are one or more date lines (denoted by DT), indicating when the entry was first created or last updated. For our sequence of interest, the entry was originally created on May 19, 1996 and was last updated in ENA on June 23, 2017:

```
DT 19-MAY-1996 (Rel. 47, Created)
DT 23-JUN-2017 (Rel. 133, Last updated, Version 5)
```

The release number in each line indicates the first quarterly release made *after* the entry was created or last updated. The version number for the entry appears on the second line and allows the user to determine easily whether they are looking at the most up-to-date record for a particular sequence. Please note that this is different from the accession.version format described above – while some element of the record may have changed, the sequence may have remained the same, so these two different types of version numbers may not always correspond to one another.

The next part of the header contains the definition lines, providing a succinct description of the kinds of biological information contained within the record. The definition line (DE in ENA, DEFINITION in DDBJ/GenBank) takes the following form.

```
DE Drosophila melanogaster eukaryotic initiation factor 4E (eIF4E) gene,
DE complete cds, alternatively spliced.
```

Much care is taken in the generation of these definition lines and, although many of them can be generated automatically from other parts of the record, they are reviewed to ensure that consistency and richness of information are maintained. Obviously, it is quite impossible to capture all of the biology underlying a sequence in a single line of text, but that wealth of information will follow soon enough in downstream parts of the same record.

Continuing down the flatfile record, one finds the full taxonomic information on the sequence of interest. The OS line (or SOURCE line in DDBJ/GenBank) provides the preferred scientific name from which the sequence was derived, followed by the common name of the organism in parentheses. The OC lines (or ORGANISM lines in DDBJ/GenBank) contain the complete taxonomic classification of the source organism. The classification is listed top-down, as nodes in a taxonomic tree, with the most general grouping (Eukaryota) given first.

```
OS Drosophila melanogaster (fruit fly)
OC Eukaryota; Metazoa; Ecdysozoa; Arthropoda; Hexapoda; Insecta; Pterygota;
OC Neoptera; Holometabola; Diptera; Brachycera; Muscomorpha; Ephydroidea;
OC Drosophilidae; Drosophila; Sophophora.
```

Each record must have at least one reference or citation, noted within what are called *reference blocks*. These reference blocks offer scientific credit and set a context explaining why this particular sequence was determined. The reference blocks take the following form.

```

RN  [1]
RP  1-2881
RX  DOI; .1074/jbc.271.27.16393.
RX  PUBMED; 8663200.
RA  Lavoie C.A., Lachance P.E., Sonenberg N., Lasko P.;
RT  "Alternatively spliced transcripts from the Drosophila eIF4E gene produce
RT  two different Cap-binding proteins";
RL  J Biol Chem 271(27):16393-16398(1996).
XX
RN  [2]
RP  1-2881
RA  Lasko P.F.;
RT  ;
RL  Submitted (09-APR-1996) to the INSDC.
RL  Paul F. Lasko, Biology, McGill University, 1205 Avenue Docteur Penfield,
RL  Montreal, QC H3A 1B1, Canada

```

In this case, two references are shown, one referring to a published paper and the other referring to the submission of the sequence record itself. In the example above, the second block provides information on the senior author of the paper listed in the first block, as well as the author's postal address. While the date shown in the second block indicates when the sequence (and accompanying information) was submitted to the database, it does not indicate when the record was first made public, so no inferences or claims based on first public release can be made based on this date. Additional submitter blocks may be added to the record each time the sequence is updated.

Some headers may contain COMMENT (DDBJ/GenBank) or CC (ENA) lines. These lines can include a great variety of notes and comments (*descriptors*) that refer to the entire record. Often, genome centers will use these lines to provide contact information and to confer acknowledgments. Comments also may include the history of the sequence. If the sequence of a particular record is updated, the comment will contain a pointer to the previous versions of the record. Alternatively, if an earlier version of the record is retrieved, the comment will point forward to the newer version, as well as backwards, if there was a still earlier version. Finally, there are database cross-reference lines (marked DR) that provide links to allied databases containing information related to the sequence of interest. Here, a cross-reference to FlyBase can be seen in the complete header for this record in Appendix 1.1. Note that the corresponding DDBJ/GenBank header in Appendix 1.2 does not contain these cross-references.

The Feature Table

Early on in the collaboration between INSDC partner organizations, an effort was made to come up with a common way to represent the biological information found within a given database record. This common representation is called the *feature table*, consisting of *feature keys* (a single word or abbreviation indicating the described biological property), *location* information denoting where the feature is located within the sequence, and additional *qualifiers* providing additional descriptive information about the feature. The online INSDC feature table documentation is extensive and describes in great detail what features are allowed and what qualifiers can be used with each individual feature. Wording within the feature table uses common biological research terminology wherever possible and is consistent between DDBJ, ENA, and GenBank entries.

Here, we will dissect the feature table for the eukaryotic transcription factor 4E gene from *Drosophila melanogaster*, shown in its entirety in both Appendices 1.3 (in ENA format) and 1.4 (in DDBJ/GenBank format). This particular sequence is alternatively spliced, producing two distinct gene products, 4E-I and 4E-II. The first block of information in the feature table is always the source feature, indicating the biological source of the sequence and additional information relating to the entire sequence. This feature must be present in all INSDC entries, as all DNA or RNA sequences derive from some specific biological source, including synthetic DNA.

```

FT   source           1..2881
FT                       /organism="Drosophila melanogaster"
FT                       /chromosome="3"
FT                       /map="67A8-B2"
FT                       /mol_type="genomic DNA"
FT                       /db_xref="taxon:7227"
FT   gene             80..2881
FT                       /gene="eIF4E"

```

In the first line of the source key, notice that the numbering scheme shows the range of positions covered by this feature key as two numbers separated by two dots (1..2881). As the source key pertains to the entire sequence, we can infer that the sequence described in this entry is 2881 nucleotides in length. The various ways in which the location of any given feature can be indicated are shown in Table 1.1, accounting for a wide range of biological scenarios. The qualifiers then follow, each preceded by a slash. The full scientific name of the organism is provided, as are specific mapping coordinates, indicating that this sequence is at map location 67A8-B2 on chromosome 3. Also indicated is the type of molecule that was sequenced (genomic DNA). Finally, the last line indicates a database cross-reference (abbreviated as db_xref) to the NCBI taxonomy database, where taxon 7227 corresponds to *D. melanogaster*. In general, these cross-references are controlled qualifiers that allow entries to be connected to an external database, using an identifier that is unique to that external database. Following the source block above is the gene feature, indicating that the gene itself is a subset of the entire sequence in this entry, starting at position 80 and ending at position 2881.

```

FT   mRNA            join(80..224,892..1458,1550..1920,1986..2085,2317..2404,
FT                       2466..2881)
FT                       /gene="eIF4E"
FT                       /product="eukaryotic initiation factor 4E-I"
FT   mRNA            join(80..224,1550..1920,1986..2085,2317..2404,2466..2881)
FT                       /gene="eIF4E"
FT                       /product="eukaryotic initiation factor 4E-II"

```

Table 1.1 Indicating locations within the feature table.

345	Single position within the sequence
345..500	A continuous range of positions bounded by and including the indicated positions
<345..500	A continuous range of positions, where the exact lower boundary is not known; the feature begins somewhere prior to position 345 but ends at position 500
345..>500	A continuous range of positions, where the exact upper boundary is not known; the feature begins at position 345 but ends somewhere after position 500
<1..888	The feature starts before the first sequenced base and continues to position 888
(102..110)	Indicates that the exact location is unknown, but that it is one of the positions between 102 and 110, inclusive
123 ^ 124	Points to a site <i>between</i> positions 123 and 124
123 ^ 177	Points to a site <i>between</i> two adjacent nucleotides or amino acids anywhere between positions 123 and 177
join(12..78,134..202)	Regions 12–78 and 134–202 are joined to form one contiguous sequence
complement(4918..5126)	The sequence complementary to that found from 4918 to 5126 in the sequence record
J00194:100..202	Positions 100–202, inclusive, in the entry in this database having accession number J00194

The next feature in this example indicates which regions form the two mRNA transcripts for this gene, the first for eukaryotic initiation factor 4E-I and the second for eukaryotic initiation factor 4E-II. In the first case (shown above), the `join` line indicates that six distinct DNA segments are transcribed to form the mature RNA transcript while, in the second case, the second region is missing, with only five distinct DNA segments transcribed into the mature RNA transcript – hence the two splice variants that are ultimately encoded by this molecule.

```

FT   CDS           join(201..224,1550..1920,1986..2085,2317..2404,2466..2629)
FT           /codon_start=1
FT           /gene="eIF4E"
FT           /product="eukaryotic initiation factor 4E-II"
FT           /note="Method: conceptual translation with partial peptide
FT           sequencing"
FT           /db_xref="GOA:P48598"
FT           /db_xref="InterPro:IPR001040"
FT           /db_xref="InterPro:IPR019770"
FT           /db_xref="InterPro:IPR023398"
FT           /db_xref="PDB:4AXG"
FT           /db_xref="PDB:4UE8"
FT           /db_xref="PDB:4UE9"
FT           /db_xref="PDB:4UEA"
FT           /db_xref="PDB:4UEB"
FT           /db_xref="PDB:4UEC"
FT           /db_xref="PDB:5ABU"
FT           /db_xref="PDB:5ABV"
FT           /db_xref="PDB:5T47"
FT           /db_xref="PDB:5T48"
FT           /db_xref="UniProtKB/Swiss-Prot:P48598"
FT           /protein_id="AAC03524.1"
FT           /translation="MVLLETEKTSAPSTEQGRPEPPTSAAAPAEAKDVKPKEDPQETGE
FT           PAGNTATTTAPAGDDAVRTEHLYKHLPLMNVWTLWYLENDRSKSWEDMQNEITSFDTVED
FT           FWSLYNHIKPPSEIKLGS DYSLFKKNIRPMWEDAANKQGRWVITLNKSSKTDLDNLWL
FT           DVLLCLIGEAPDHS DQICGAVINIRGKSNKIS IWTADGNNEAALEIGHKLRDALRLGR
FT           NNSLQYQLHKDTMVKQGSNVKSIYTL"

```

Following the mRNA feature is the CDS feature shown above, describing the region that ultimately encodes the protein product. Focusing just on eukaryotic initiation factor 4E-II, the CDS feature also shows a `join` line with coordinates that are slightly different from those shown in the mRNA feature, specifically at the beginning and end positions. The difference lies in the fact that the 5' and 3' untranslated regions (UTRs) are included in the mRNA feature but not in the CDS feature. The CDS feature corresponds to the sequence of amino acids found in the translated protein product whose sequence is shown in the `/translation` qualifier above. The `/codon_start` qualifier indicates that the amino acid translation of the first codon begins at the first position of this joined region, with no offset.

The `/protein_id` qualifier shows the accession number for the corresponding entry in the protein databases (AAC03524.1) and is hyperlinked, enabling the user to go directly to that entry. These unique identifiers use a “3 + 5” format – three letters, followed by five numbers. Versions are indicated by the decimal that follows; when the protein sequence in the record changes, the version is incremented by one. The assignment of a gene product or protein name (via the `/protein` qualifier) often is subjective, sometimes being assigned via weak similarities to other (and sometimes poorly annotated) sequences. Given the potential for the transitive propagation of poor annotations (that is, bad data tend to beget more bad data), users are advised to consult *curated* nucleotide and protein sequence databases for the most up-to-date, accurate information regarding the putative function of a given sequence. Finally, notice the extensive cross-referencing via the `/db_xref` qualifier to entries in InterPro, the

Protein Data Bank (PDB), and UniProtKB/Swiss-Prot, as well as to a Gene Ontology annotation (GOA; Gene Ontology Consortium 2017).

Implicit in the source feature and the organism that is assigned to it is the genetic code used to translate the nucleic acid sequence into a protein sequence when a CDS feature is present in the record. Also, the DNA-centric nature of these feature tables means that all features are mapped through a DNA coordinate system, not that of amino acid reference points, as shown in the examples in Appendices 1.3 and 1.4.

```
SQ   Sequence 2881 BP; 849 A; 699 C; 585 G; 748 T; 0 other;
      cggttgcttg ggtttataa catcagtcag tgacaggcat ttccagagtt gccctgttca      60
      acaatcgata gctgcctttg gccacaaaa tcccaactt aattaaagaa ttaaataatt      120
      cgaataataa ttaagccag taacctacgc agcttgagtg cgtaaccgat atctagtata      180
      .
      . <truncated for brevity>
      .
      aaacggaacc ccctttgta tcaaaaatcg gcataatata aaatctatcc gctttttgta      2820
      gtcactgtca ataatggatt agacggaaaa gtatattaat aaaaacctac attaaaaccg      2880
      g                                                    2881
//
```

Finally, at the end of every nucleotide sequence record, one finds the actual nucleotide sequence, with 60 bases per row. Note that, in the SQ line signaling the beginning of this section of the record, not only is the overall length of the sequence provided, but a count of how many of each individual type of nucleotide base is also provided, making it quite easy to compute the GC content of this sequence.

Graphical Interfaces

Graphical interfaces have been developed to facilitate the interpretation of the data found within text-based flatfiles, with an example of the graphical view of the ENA record for our sequence of interest (U54469.1) shown in Figure 1.1. These graphical views are particularly useful when there is a long list of documented biological features within the feature table, enabling the user to visualize potential interactions or relationships between biological features. An additional example of the use of graphical views to assist in the interpretation of the information found within a database record is provided in the discussion of the NCBI Entrez discovery pathway in Chapter 2, as well as later in this chapter.

RefSeq

As one might expect, especially given the breakneck speed at which DNA sequence data are currently being produced, there is a significant amount of redundancy within the major sequence databases, with a good number of sequences being represented more than once. This is often problematic for the end user, who may find themselves confused as to which sequence to use after performing a search that returns numerous results. To address this issue, NCBI developed RefSeq, the goal of which is to provide a single reference sequence for each molecule of the central dogma – DNA, RNA, and protein. The distinguishing features of RefSeq go beyond its non-redundant nature, with individual entries including the biological attributes of the gene, gene transcript, or protein. RefSeq entries encompass a wide taxonomic range, and entries are updated and curated on an ongoing basis to reflect current knowledge about the individual entries. Additional information on RefSeq can be found in Box 1.2.

The screenshot shows the ENA record landing page for U54469.1. The page title is "Sequence: U54469.1" and the description is "Drosophila melanogaster eukaryotic initiation factor 4E (eIF4E) gene, complete cds, alternatively spliced." The page includes navigation tabs for "Navigation", "Overview", "Source Feature(s)", "Sequence", "Publications", "Submission Details", and "Other Feature(s)". The "Overview" tab is selected, showing a graphical view of biological features. The graphical view displays the forward strand of the 2,881 bp sequence. The features shown are:

- Source: Drosophila melanogaster
- Genes: eIF4E
- mRNA: eIF4E
- CDS: eIF4E

The graphical view also shows the position of the gene, mRNAs, and coding regions (marked CDS) within the 2,881 bp sequence. The "Base range" is set to 1 - 2881.

Figure 1.1 The landing page for ENA record U54469.1, providing a graphical view of biological features found within the sequence of the *Drosophila melanogaster* eukaryotic initiation factor 4E (*eIF4E*) gene. The tracks within the graphical view show the position of the gene, mRNAs, and coding regions (marked CDS) within the 2881 bp sequence reported in this record.

Box 1.2 RefSeq

The first several chapters of this book describe a variety of ways in which sequence data and sequence annotations find their way into public databases. While the combination of data derived from systematic sequencing projects and individual investigators' laboratories yields a rich and highly valuable set of sequence data, some problems are apparent. The most important issue is that a single biological entity may be represented by many different entries in various databases. It also may not be clear whether a given sequence has been experimentally determined or is simply the result of a computational prediction.

To address these issues, NCBI developed the RefSeq project, the major goal of which is to provide a reference sequence for each molecule in the central dogma (DNA, mRNA, and protein). As each biological entity is represented only once, RefSeq is, by definition, non-redundant. Nucleotide and protein sequences in RefSeq are explicitly linked to one

another. Most importantly, RefSeq entries undergo ongoing curation, assuring that the RefSeq entry represents the most up-to-date state of knowledge regarding a particular DNA, mRNA, or protein sequence.

RefSeq entries are distinguished from other entries in GenBank through the use of a distinct accession number series. RefSeq accession numbers follow a “2 + 6” format: a two-letter code indicating the type of reference sequence, followed by an underscore and a six-digit number. Experimentally determined sequence data are denoted as follows:

NT_123456	Genomic contigs (DNA)
NM_123456	mRNAs
NP_123456	Proteins

Reference sequences derived through genome annotation efforts are denoted as follows:

XM_123456	Model mRNAs
XP_123456	Model proteins

It is important to understand the distinction between the “N” numbers and “X” numbers – the former represent actual, experimentally determined sequences, while the latter represent computational predictions derived from the raw DNA sequence.

Additional types of RefSeq entries, along with more information on the RefSeq project, can be found on the NCBI RefSeq web site.

Protein Sequence Databases

With the availability of myriad complete genome sequences from both prokaryotes and eukaryotes, significant effort is being dedicated to the identification and functional analysis of the proteins encoded by these genomes. The large-scale analysis of these proteins continues to generate huge amounts of data, including through the use of proteomic methods (Chapter 11) and through protein structure analysis (Chapter 12), to name a few. These and other methods make it possible to identify large numbers of proteins quickly, to map their interactions (Chapter 13), to determine their location within the cell, and to analyze their biological activities. This ever-increasing “information space” reinforces the central role that protein sequence databases play as a resource for storing data generated by these efforts, making them freely available to the life sciences community.

As most sequence data in protein databases are derived from the translation of nucleotide sequences, they can be, in large part, thought of as “secondary databases.” Universal protein sequence databases cover proteins from all species, whereas specialized protein sequence databases concentrate on particular protein families, groups of proteins, or those from a specific organism. Representative model organism databases include the Mouse Genome Database (MGD; Smith et al. 2018) and WormBase (Lee et al. 2018), among others (Baxevanis and Bateman 2015; Rigden and Fernández 2018). Organismal sequence databases are discussed in greater detail in Chapter 2.

Universal protein databases can be divided further into two broad categories: sequence repositories, where the data are stored with little or no manual intervention, and curated databases, in which experts enhance the original data through expert *biocuration*. The importance of ensuring interoperability, creating and implementing standards, and adopting best practices aimed at accurately representing the biological knowledge found within the sequence databases is absolutely paramount. Indeed, these curation goals are so important that there is an organization called the International Society for Biocuration, the primary mission of which is to advance these central tenets.

The NCBI Protein Database

NCBI maintains the Protein database, which derives its content from a number of sources. These include the translations of the annotated coding regions from INSDC databases described above, from RefSeq (Box 1.2), and from NCBI's Third Party Annotation (TPA) database. The TPA dataset is quite interesting in its own right, as it captures both experimental and inferential data provided by the scientific community to supplement the information found in an INSDC nucleotide entry. As the name suggests, the information in the TPA is provided by third parties and not by the original submitter of the corresponding INSDC entry. The NCBI Protein database also includes additional non-NCBI sources of protein sequence data, including Swiss-Prot, PIR, PDB, and the Protein Research Foundation. Step-by-step methods for performing searches against the NCBI Protein database are described in detail in Chapter 3.

UniProt

Although data repositories are an essential vehicle through which scientists can access sequence data as quickly as possible, it is clear that the addition of biological information from

UniProtKB - P09651 (ROA1_HUMAN)

Display: BLAST, Align, Format, Add to basket, History

Entry: **Protein** Heterogeneous nuclear ribonucleoprotein A1
Gene HNRNPA1
Organism *Homo sapiens (Human)*
Status Reviewed - Annotation score: ★★★★★ - Experimental evidence at protein levelⁱ

Functionⁱ
Involved in the packaging of pre-mRNA into hnRNP particles, transport of poly(A) mRNA from the nucleus to the cytoplasm and may modulate splice site selection (PubMed:17371836). May bind to specific miRNA hairpins (PubMed:28431233). ² Publications
(Microbial infection) May play a role in HCV RNA replication. ¹ Publication

Caution
Variant Val-314 has been originally associated with IBMPFD3 and variant Asn-314 with ALS20 (PubMed:25616961). However in another report, variant Val-314 is associated with amyotrophic lateral sclerosis (ALS) but this variant is not supported by clinical data in this publication (PubMed:25616961). ¹ Curated

GO - Molecular functionⁱ

- G-rich strand telomeric DNA binding ¹ Source: BHF-UCL
- Identical protein binding ¹ Source: IntAct
- miRNA binding ¹ Source: UniProtKB
- pre-mRNA binding ¹ Source: CAFA
- protein domain specific binding ¹ Source: UniProtKB
- RNA binding ¹ Source: UniProtKB
- single-stranded DNA binding ¹ Source: HGNC
- single-stranded RNA binding ¹ Source: HGNC
- telomeric repeat-containing RNA binding ¹ Source: BHF-UCL

View the complete GO annotation on QuickGO ...

GO - Biological processⁱ

- cellular response to glucose starvation ¹ Source: CAFA
- cellular response to sodium arsenite ¹ Source: UniProtKB
- fibroblast growth factor receptor signaling pathway ¹ Source: Reactome
- import into nucleus ¹ Source: HGNC
- mRNA splicing, via spliceosome ¹ Source: Reactome
- mRNA transport ¹ Source: UniProtKB-KW
- negative regulation of telomere maintenance via telomerase ¹ Source: BHF-UCL
- nuclear export ¹ Source: HGNC
- positive regulation of telomere maintenance via telomerase ¹ Source: BHF-UCL
- regulation of alternative mRNA splicing, via spliceosome ¹ Source: CAFA
- RNA export from nucleus ¹ Source: HGNC
- RNA metabolic process ¹ Source: Reactome
- viral process ¹ Source: UniProtKB-KW

View the complete GO annotation on QuickGO ...

Keywordsⁱ

Molecular function: Ribonucleoprotein, RNA-binding

Biological process: Host-virus interaction, mRNA processing, mRNA splicing, mRNA transport, Transport

Enzyme and pathway databases

Reactome¹
R-HSA-6803529 FGFR2 alternative splicing
R-HSA-72163 mRNA Splicing - Major Pathway
R-HSA-72203 Processing of Capped Intron-Containing Pre-mRNA

Figure 1.2 Results of a search for the human heterogeneous nuclear ribosomal protein A1 record within UniProtKB, using the accession number P09651 as the search term. See text for details.

multiple, highly regarded sources greatly increases the power of the underlying sequence data. The UniProt Consortium was formed to accomplish just that, bringing together the Swiss-Prot, TrEMBL, and the Protein Information Resource Protein Sequence Database under a single umbrella, called UniProt (UniProt Consortium 2017). UniProt comprises three main databases: the UniProt Archive, a non-redundant set of all publicly available protein sequences compiled from a variety of source databases; UniProtKB, combining entries from UniProtKB/Swiss-Prot and UniProtKB/TrEMBL; and the UniProt Reference Clusters (UniRef), containing non-redundant views of the data contained in UniParc and UniProtKB that are clustered at three different levels of sequence identity (Suzek et al. 2015).

The wealth of information found within a UniProtKB entry can be best illustrated by an example. Here, we will consider the entry for the human heterogeneous nuclear ribonuclear protein A1, with accession number P09651. A search of UniProtKB using this accession number as the search term produces the view seen in Figure 1.2. The lower part of the left-hand column shows the various types of information available for this protein, and the user can select or de-select sections based on their interests. The main part of the window provides basic

Subcellular location¹

UniProt annotation: GO - Cellular component

Nucleus

- Nucleus ⓘ # 2 Publications -

Other locations

Cytoplasm ⓘ # 1 Publication -

Note: Localized in cytoplasmic mRNA granules containing untranslated mRNAs. Shuttles continuously between the nucleus and the cytoplasm along with mRNA. Component of ribonucleosomes (PubMed:17289663). # 2 Publications -

Other locations

Cytoplasm ⓘ # 1 Publication -

Note: (Microbial infection) In the course of viral infection, colocalizes with HCV NS5B at speckles in the cytoplasm in a HCV-replication dependent manner. # 1 Publication -

Keywords - Cellular component¹
Cytoplasm, Nucleus, Spliceosome

Pathology & Biotech¹

Involvement in disease¹

Inclusion body myopathy with early-onset Paget disease with or without frontotemporal dementia 3 (IBMPFD3) ⓘ # 1 Publication -

The disease is caused by mutations affecting the gene represented in this entry.
Disease description: An autosomal dominant disease characterized by disabling muscle weakness clinically resembling to limb girdle muscular dystrophy, osteolytic bone lesions consistent with Paget disease, and premature frontotemporal dementia. Clinical features show incomplete penetrance.
See also OMIM:615424

Feature key	Position(s)	Description	Actions	Graphical view	Length
Natural variant ¹ (VAR_070588)	314	D → V in IBMPFD3; reduces binding to UBQLN2. dbSNP:rs397518452	ⓘ # 2 Publications - Corresponds to variant Ensembl, ClinVar.		1

Amyotrophic lateral sclerosis 20 (ALS20) ⓘ # 2 Publications -

The disease is caused by mutations affecting the gene represented in this entry.
Disease description: A neurodegenerative disorder affecting upper motor neurons in the brain and lower motor neurons in the brain stem and spinal cord, resulting in fatal paralysis. Sensory abnormalities are absent. The pathologic hallmarks of the disease include pallor of the corticospinal tract due to loss of motor neurons, presence of ubiquitin-positive inclusions within surviving motor neurons, and deposition of pathologic aggregates. The etiology of amyotrophic lateral sclerosis is likely to be multifactorial, involving both genetic and environmental factors. The disease is inherited in 5-10% of the cases.
See also OMIM:615426

Feature key	Position(s)	Description	Actions	Graphical view	Length
Natural variant ¹ (VAR_077631)	277	Q → K in ALS20; unknown pathological significance. ⓘ # 1 Publication -			1
Natural variant ¹ (VAR_070588)	314	D → N in ALS20. ⓘ # 1 Publication - Corresponds to variant dbSNP:rs397518453	Ensembl, ClinVar.		1
Natural variant ¹ (VAR_070590)	319	N → S in ALS20. ⓘ # 1 Publication - Corresponds to variant dbSNP:rs397518454	Ensembl, ClinVar.		1
Natural variant ¹ (VAR_077633)	340	P → S in ALS20; increases subcellular localization of HNRNPA1 in cytoplasmic inclusions with stress granules. ⓘ # 1 Publication -			1

Mutagenesis

Feature key	Position(s)	Description	Actions	Graphical view	Length
Mutagenesis ¹	326	G → A: No nuclear import nor export. ⓘ # 1 Publication -			1
Mutagenesis ¹	327	P → A: No nuclear import nor export. ⓘ # 1 Publication -			1
Mutagenesis ¹	334 - 335	GG → LL: Normal nuclear import and export. ⓘ # 1 Publication -			2

Figure 1.3 The Subcellular location and Pathology & Biotech sections of the record for the human heterogeneous nuclear ribosomal protein A1 record within UniProtKB. These sections can be accessed by clicking on the blue tiles in the left-hand column of the window. See text for details.

identifying information about this sequence, as well as an indication of whether the entry has been manually reviewed and annotated by UniProtKB curators. Here, we see that the entry has indeed been reviewed and that there is experimental evidence that supports the existence of the protein. The next section in the file is devoted to conveying functional information, also providing Gene Ontology (GO) terms that are associated with the entry, as well as links to enzyme and pathway databases such as Reactome (see Chapter 13). Clicking on any of the blue tiles in the left-hand column will jump the user down to the selected section of the entry. For instance, if one clicks on Subcellular location, the view seen in Figure 1.3 is produced, providing a color-coded schematic of the cell indicating the type of annotation (manual or automatic) and links to publications supporting the annotation. The lower part of Figure 1.3 also shows information regarding the protein's involvement in disease, documenting variants that have been implicated in early onset Paget disease and amyotrophic lateral sclerosis (Kim et al. 2013; Liu et al. 2016).

In the upper left corner of the UniProtKB window are display options that are quite useful in visualizing the significant amount of data found in this entry's feature table. By clicking on Feature viewer, one is presented with the view shown in Figure 1.4, neatly summarizing

The screenshot displays the UniProtKB Feature viewer for the protein P09651 (ROA1_HUMAN). The main view shows a sequence map from position 1 to 372. A pop-up window titled 'MOD_RES 365' provides details for a phosphoserine modification at position 365, including its description and evidence from PubMed and EuropePMC. Below the sequence map, a 3D protein structure is shown, and a table lists PDB entries for the protein.

PDB Entry	Method	Resolution	Chain	Positions	Links
1MA1	X-ray	1.75 Å	A	1-184	PDBe RCSB P... PDBJ PDBsum
1L3K	X-ray	1.10 Å	A	1-196	PDBe RCSB P... PDBJ PDBsum
1PGZ	X-ray	2.60 Å	A	2-196	PDBe RCSB P... PDBJ PDBsum
1P06	X-ray	2.10 Å	A	8-190	PDBe RCSB P... PDBJ PDBsum
1U1K	X-ray	2.00 Å	A	1-196	PDBe RCSB P... PDBJ PDBsum
1U1L	X-ray	2.00 Å	A	1-196	PDBe

Figure 1.4 The Feature viewer rendering of the record for the human heterogeneous nuclear ribosomal protein A1 within UniProtKB. Clicking the Display link, found in the upper left portion of the window, provides access to the Feature viewer. Any of the sections can be expanded by clicking on the labels in the blue boxes to the left of the graphic. See text for details.

the annotations for this sequence in a coordinate-based fashion. Any of the sections can be expanded by clicking on the labels in the blue boxes to the left of the graphic. Here, the post-translational modification (PTM) section has been expanded, showing the position of modified residues in this protein; clicking on any of the markers in the track will produce a pop-up with additional information on the PTM, along with relevant links to the literature. In Figure 1.5, the Structural features and Variants sections have also been expanded, showing the positions of all alpha helices, beta strands, and beta turns within the protein, as well as the location of putatively clinically relevant point mutations. Here, a variant at position 351 is highlighted, with the proline-to-leucine variant identified as part of the ClinVar project (Lan-drum et al. 2016) having a possible association with relapsing–remitting multiple sclerosis. By examining different sections of this very useful graphical display, the user can start to see how various features overlap with one another, perhaps indicating whether a known or predicted disease-causing variant falls within a structured region of the protein. These annotations and observations can provide important insights with respect to experimental design and the interpretation of experimental data.

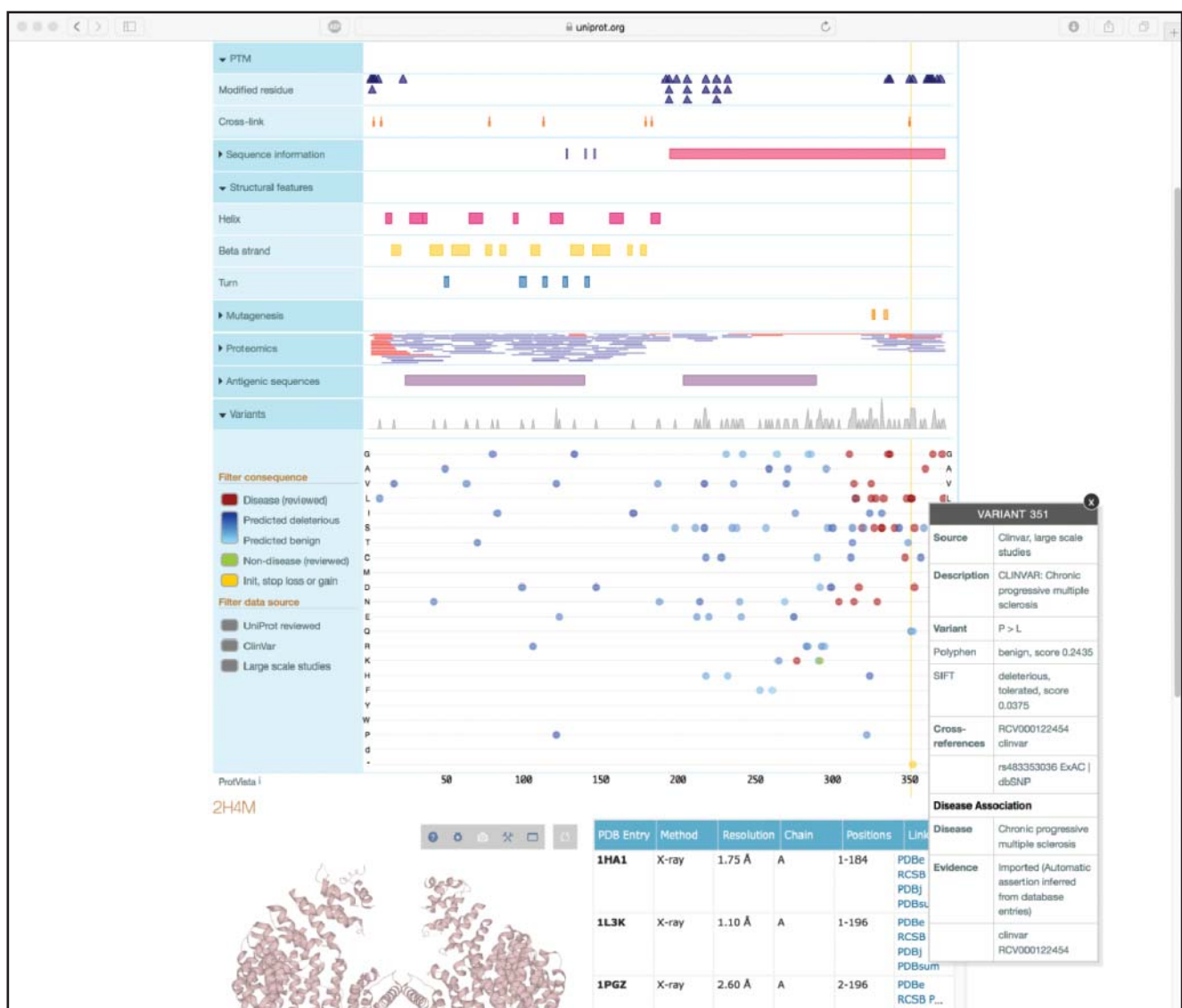


Figure 1.5 Expanding the PTM, Structural features, and Variants sections within the Feature viewer display shows the position of all post-translational modifications (PTMs), alpha helices, beta strands, and beta turns within the human heterogeneous nuclear ribosomal protein A1, as well as the location of putatively clinically relevant point mutations. Clicking on any of the variants produces a pop-up window with additional information; here, the pop-up window provides disease association data for the proline-to-leucine variant at position 351 of the sequence. See text for details.

Summary

The rapid pace of discovery in the genomic and proteomic arenas requires that databases are built in a way that facilitates not just the storage of these data, but the efficient handling and retrieval of information from these databases. Many lessons have been learned over the decades regarding how to approach critical questions regarding design and content, often the hard way. Thus, the continued development of currently existing databases, as well as the conceptualization and creation of new types of databases, will be a critical focal point for the advancement of biological discovery. As should be obvious from this chapter, keeping databases up to date and accurate is a task that requires the active involvement of the biological community (Box 1.3). Therefore, it is incumbent upon all users to ensure the accuracy of these data in an active fashion, engaging the curators in a continuous dialog so that these widely used resources continue to remain a valuable resource to biologists worldwide.

Box 1.3 Ensuring the Continued Quality of Data in Public Sequence Databases

Given the roles of DDBJ, EMBL, and GenBank in maintaining the archive of all publicly available DNA, RNA, and protein sequences, the continued usefulness of this resource is highly dependent on the quality of data found within it. Despite the high degree of both manual and automated checking that takes place before a record becomes public, errors will still find their way into the databases. These errors may be trivial and have no biological consequence (e.g. an incorrect postal code), may be misleading (e.g. an organism having the correct genus but wrong species name), or downright incorrect (e.g. a full-length mRNA not having a CDS annotated on it). Sometimes, records may have incorrect reference blocks, preventing researchers from linking to the correct publication describing the sequence. Over time, many have taken an active role in reporting these errors but, more often than not, these errors are left uncorrected.

While the individual INSDC members have the responsibility for hosting and disseminating the data found within their databases, keep in mind that the ownership of the data rests with the original submitter – and these original submitters (or their designees) are the only ones who can make updates to their database records. To keep these community resources as accurate and up to date as possible, users are actively encouraged to report any errors found when using the databases in the course of their work so that the database administrators can follow up with the original submitters as appropriate.

Given below are the current e-mail addresses for submitting information regarding errors to the three major sequence databases. As all the databases share information with each other nightly, it is only necessary to report the error to one of the three members of the consortium. Authors are actively encouraged to check their own records periodically to ensure that the information they previously submitted is still accurate. Even though this charge to the community is discussed here in the context of the three major sequence databases, all databases provide similar mechanisms through which incorrect information can be brought to the attention of the database administrators.

DDBJ	ddbjudpt@dbj.nig.ac.jp
EMBL	datasubs@ebi.ac.uk
GenBank	gb-admin@ncbi.nlm.nih.gov

As alluded to above, the range of publicly available data obviously goes well beyond human data, whether sequence based or not. As the major public sequence databases need to be able to store data in a fairly generalized fashion, these databases often do not contain more specialized types of information that would be of interest to specific segments of the biological community. To address this, many smaller, specialized databases have emerged and have been developed and curated by biologists “in the trenches” to fulfill specific needs. These databases, which contain information ranging from strain crosses to gene expression data, provide a valuable

adjunct to the more visible public sequence databases, and users are encouraged to make intelligent use of both types of databases. An annotated list of such databases can be found in the yearly Database issue of *Nucleic Acids Research* (Rigden and Fernández 2018).

The position of this chapter at the beginning of this book reflects the belief that an understanding of biological databases is the first step toward being able to perform robust and accurate bioinformatic analyses. The reader is very strongly encouraged to take the time to understand the structure of the data found within these databases, as the basis for finding sequence data of interest and performing the more advanced analyses described in the chapters that follow.

Acknowledgments

The author thanks Rolf Apweiler for the use of material from the third edition of this book.

Internet Resources

DDBJ Database Divisions	www.ddbj.nig.ac.jp/ddbj/data-categories-e.html
DNA Database of Japan (DDBJ)	www.ddbj.nig.ac.jp
EMBL Nucleotide Sequence Database	www.embl.org
ENA Data Formats	www.ebi.ac.uk/ena/submit/data-formats
European Bioinformatics Institute	www.ebi.ac.uk
GenBank	www.ncbi.nlm.nih.gov
GenBank Database Divisions	www.ncbi.nlm.nih.gov/genbank/htgs/divisions
Genome Ontology Consortium	geneontology.org
INSDC Feature Table Definition	insdc.org/documents/feature_table.html
International Society for Biocuration	biocuration.org
NCBI Data Model	www.ncbi.nlm.nih.gov/IEB/ToolBox/SDKDOCS/DATAMODL.HTML
NCBI Protein Database	www.ncbi.nlm.nih.gov/protein
<i>Nucleic Acids Research</i> Database issue	academic.oup.com/nar
Protein Data Bank (PDB)	www.rcsb.org
Protein Identification Resource (PIR)	pir.georgetown.edu
Protein Research Foundation	www.proteinresearch.net
RefSeq	www.ncbi.nlm.nih.gov/refseq
Swiss-Prot (EBI)	www.ebi.ac.uk/uniprot
Swiss-Prot (ExPASy)	web.expasy.org/docs/swiss-prot_guideline.html
UniProt Consortium	www.uniprot.org

Further Reading

- Bairoch, A. (2000). Serendipity in bioinformatics: the tribulations of a Swiss bioinformatician through exciting times! *Bioinformatics*. 16: 48–64. A personal narrative conveying the early history of the development of sequence databases and related software tools, events that set the groundwork for the modern bioinformatics landscape.
- Green, E.D., Rubin, E.M., and Olson, M.V. (2017). The future of DNA sequencing. *Nature*. 550: 179–181. An insightful perspective regarding the next several decades of the application of DNA sequencing methodologies in novel contexts and the implications of those applications to issues of data storage and data sharing.
- Rigden, D.J. and Fernández, X.M. (2018). The 2018 *Nucleic Acids Research* database issue and the online molecular biology database collection. *Nucleic Acids Res.* 46: D1–D7. The 25th overview

of the annual database issue published by *Nucleic Acids Research*, capturing the wide variety of publicly available bioinformatic databases available to the community. This overview is updated yearly, and the individual papers describing these database resources are freely available through the *Nucleic Acids Research* web site.

References

- Apweiler, R. (2001). Functional information in Swiss-Prot: the basis for large-scale characterization of protein sequences. *Briefings Bioinf.* 2: 9–18.
- Bairoch, A. (2000). Serendipity in bioinformatics: the tribulations of a Swiss bioinformatician through exciting times! *Bioinformatics.* 16: 48–64.
- Baxevanis, A.D. and Bateman, A. (2015). The importance of biological databases in biological discovery. *Curr. Protoc. Bioinf.* 50: 1.1.1–1.1.8.
- Benson, D.A., Cavanaugh, M., Clark, K. et al. (2018). GenBank. *Nucleic Acids Res.* 46: D41–D47.
- Cook, C.E., Bergman, M.T., Cochrane, G. et al. (2018). The European Bioinformatics Institute in 2017: data coordination and integration. *Nucleic Acids Res.* 46: D21–D29.
- Dayhoff, M.O., Eck, R.V., Chang, M.A., and Sochard, M.R. (1965). *Atlas of Protein Sequence and Structure*. Silver Spring, MD: National Biomedical Research Foundation.
- Gene Ontology Consortium (2017). Expansion of the Gene Ontology knowledgebase and resources. *Nucleic Acids Res.* 45: D331–D338.
- Green, E.D., Rubin, E.M., and Olson, M.V. (2017). The future of DNA sequencing. *Nature.* 550: 179–181.
- Karsch-Mizrachi, I., Tagaki, T., and Cochrane, G., on behalf of the International Nucleotide Sequence Database Collaboration (2018). The International Nucleotide Sequence Database Collaboration. *Nucleic Acids Res.* 46: D48–D51.
- Kim, H.J., Kim, N.C., Wang, Y.D. et al. (2013). Mutations in prion-like domains in hnRNPA2B1 and hnRNPA1 cause multisystem proteinopathy and ALS. *Nature.* 495: 467–473.
- Kodama, Y., Mashima, J., Kosuge, T. et al. (2018). DNA Data Bank of Japan: 30th anniversary. *Nucleic Acids Res.* 46: D30–D35.
- Landrum, M.J., Lee, J.M., Benson, M. et al. (2016). ClinVar: public archive of interpretations of clinically relevant variants. *Nucleic Acids Res.* 44: D862–D868.
- Lee, R.Y.N., Howe, K.L., Harris, T.W. et al. (2018). WormBase 2017: molting into a new stage. *Nucleic Acids Res.* 46: D869–D874.
- Lipman, D.J. and Pearson, W.R. (1985). Rapid and sensitive protein similarity searches. *Science.* 227: 1435–1441.
- Liu, Q., Shu, S., Wang, R.R. et al. (2016). Whole-exome sequencing identifies a missense mutation in hnRNPA1 in a family with flail arm ALS. *Neurology.* 87: 1763–1769.
- Rigden, D.J. and Fernández, X.M. (2018). The 2018 *Nucleic Acids Research* database issue and the online molecular biology database collection. *Nucleic Acids Res.* 46: D1–D7.
- Silvester, N., Alako, B., Amid, C. et al. (2018). The European Nucleotide Archive in 2017. *Nucleic Acids Res.* 46: D36–D40.
- Smith, C.L., Blake, J.A., Kadin, J.A. et al., and The Mouse Genome Database Group (2018). Mouse Genome Database (MGD)-2018: knowledgebase for the laboratory mouse. *Nucleic Acids Res.* 46: D836–D842.
- Suzek, B.E., Wang, Y., Huang, H. et al., and The UniProt Consortium (2015). UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics.* 31: 926–932.
- UniProt Consortium (2017). UniProt: the universal protein knowledgebase. *Nucleic Acids Res.* 45: D158–D169.

This chapter was written by Dr. Andreas D. Baxevanis in his private capacity. No official support or endorsement by the National Institutes of Health or the United States Department of Health and Human Services is intended or should be inferred.

2

Information Retrieval from Biological Databases

Andreas D. Baxevanis

Introduction

On April 14, 2003, the biological community celebrated the achievement of the Human Genome Project's major goal: the complete, accurate, and high-quality sequencing of the human genome (International Human Genome Sequencing Consortium 2001; Schmutz et al. 2004). The attainment of this goal, which many have compared to landing a person on the moon, has had a profound effect on how biological and biomedical research is conducted and will undoubtedly continue to have a profound effect on its direction in the future. The availability of not just human genome data, but also human sequence variation data, model organism sequence data, and information on gene structure and function provides fertile ground for biologists to better design and interpret their experiments in the laboratory, fulfilling the promise of bioinformatics in advancing and accelerating biological discovery.

One of the most important databases available to biologists is GenBank, the annotated collection of all publicly available DNA and protein sequences (Benson et al. 2017; see Chapter 1). This database, maintained by the National Center for Biotechnology Information (NCBI) at the National Institutes of Health (NIH), represents a collaborative effort between NCBI, the European Molecular Biology Laboratory (EMBL), and the DNA Data Bank of Japan (DDBJ). At the time of this writing, GenBank contained over 200 million sequences and over 300 trillion nucleotide bases. The completion of human genome sequencing and the sequencing of an ever-expanding number of model organism genomes, as well as the existence of a gargantuan number of sequences in general, provides a golden opportunity for biological scientists, owing to the inherent value of these data. However, at the same time, the sheer magnitude of data presents a conundrum to the inexperienced user, resulting not just from the size of the “sequence information space” but from the fact that the information space continues to get larger and larger – by leaps and bounds – at a pace that will continue to accelerate, even though human genome sequencing has long been “completed.”

The effect of the Human Genome Project and other systematic sequencing projects on the continued accumulation of sequence data is illustrated by the growth of GenBank, as shown in Figure 2.1; the exponential growth rate illustrated in the figure is expected to continue for some time to come. The continued expansion of not just the sequence space but of the myriad biological data now available *because* of the expansion of the sequence space underscores the necessity for all biologists to learn how to effectively navigate this information for effective use in their work – even allowing investigators to *avoid* performing expensive experiments themselves based on the data found within these virtual treasure troves.

GenBank (or any other biological database, for that matter) serves little purpose unless the data can be easily searched and entries retrievable in a useful, meaningful format. Otherwise, sequencing efforts such as those described above have no useful end – without effective search and retrieval tools, the biological community as a whole cannot make use of the information hidden within these millions of bases and amino acids, much less the structures they form or

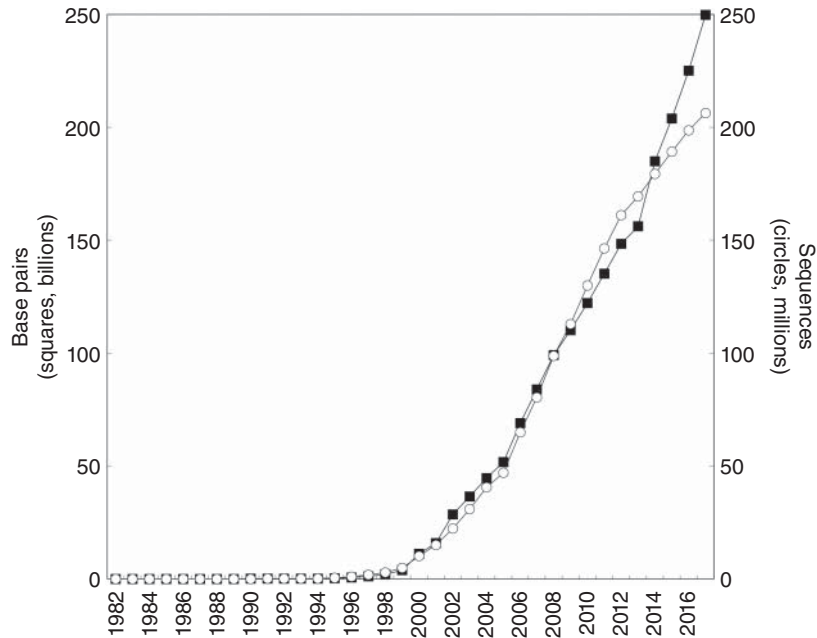


Figure 2.1 The exponential growth of GenBank in terms of number of nucleotides (squares, in millions) and number of sequences submitted (circles, in thousands). Source data for the figure have been obtained from the National Center for Biotechnology Information (NCBI) web site. Note that the period of accelerated growth after 1997 coincides with the completion of the Human Genome Project's genetic and physical mapping goals, setting the stage for high-accuracy, high-throughput sequencing, as well as the development of new sequencing technologies (Collins et al. 1998, 2003; Green et al. 2011).

the mutations they harbor. Much effort has gone into making such data accessible to the biologist, and a selection of the programs and interfaces resulting from these efforts are the focus of this chapter. The discussion will center on querying databases maintained by NCBI, as these more “general” repositories are far and away the ones most often accessed by biologists, but attention will also be given to specialized databases that provide information not necessarily found through the use of Entrez, NCBI's integrated information retrieval system.

Integrated Information Retrieval: The Entrez System

One of the most widely used interfaces for the retrieval of information from biological databases is the NCBI Entrez system. Entrez capitalizes on the fact that there are pre-existing, logical relationships between the individual entries found in numerous public databases. For example, a paper in PubMed may describe the sequencing of a gene whose sequence appears in GenBank. The nucleotide sequence, in turn, may code for a protein product whose sequence is stored in NCBI's Protein database. The three-dimensional structure of that protein may be known, and the coordinates for that structure may appear in NCBI's Structure database. Finally, there may be allelic or structural variants documented for the gene of interest, cataloged in databases such as the Single Nucleotide Polymorphism Database (called dbSNP) or the Database of Genomic Structural Variation (called dbVAR), respectively. The existence of such natural connections, all having a biological underpinning, motivated the development of a method through which all of the information about a particular biological entity could be found without having to sequentially visit and query individual databases, one by one.

Entrez, to be clear, is not a database itself. Rather, it is the interface through which its component databases can be accessed and traversed – an integrated information retrieval system. The Entrez information space includes PubMed records, nucleotide and protein sequence data, information on conserved protein domains, three-dimensional structure information, and genomic variation data with potential clinical relevance, a good number of which will be touched upon in this chapter. The strength of Entrez lies in the fact that *all* of this information, across a large number of component databases, can be accessed by issuing one – and only

one – query. This very powerful, integrated approach is made possible through the use of two general types of connections between database entries: *neighboring* and *hard links*.

Relationships Between Database Entries: Neighboring

The concept of neighboring enables entries *within* a given database to be connected to one another. If a user is looking at a particular PubMed entry, the user can then “ask” Entrez to find all of the other papers in PubMed that are similar in subject matter to the original paper. Likewise, if a user is looking at a sequence entry, Entrez can return a list of all other sequences that bear similarity to the original sequence. The establishment of neighboring relationships within a database is based on statistical measures of similarity, some of which are described in more detail below. While the term “neighboring” has traditionally been used to describe these connections, the terminology on the NCBI web site denotes neighbors as “related data.”

BLAST Biological sequence similarities are detected and sequence data are compared with one another using the Basic Local Alignment Search Tool, or BLAST (Altschul et al. 1990). This algorithm attempts to find high-scoring segment pairs – pairs of sequences that can be aligned with one another and, when aligned, meet certain scoring and statistical criteria. Chapter 3 discusses the family of BLAST algorithms and their application at length.

VAST Molecular structure similarities are detected and sets of coordinate data are compared using a vector-based method known as VAST (the Vector Alignment Search Tool; Gibrat et al. 1996). This methodology uses geometric criteria to assess similarity between three-dimensional domains, and there are three major steps that take place in the course of a VAST comparison:

- First, based on *known* three-dimensional coordinate data, the alpha helices and beta strands that constitute the structural core of each protein are identified. Straight-line vectors are then calculated based on the position of these secondary structural elements. VAST keeps track of how one vector is connected to the next (that is, how the C-terminal end of one vector connects to the N-terminal end of the next vector), as well as whether each vector represents an alpha helix or a beta strand. Subsequent comparison steps use *only* these vectors in assessing structural similarity to other proteins – so, in effect, most of the painstakingly deduced atomic coordinate data are discarded at this step. The reason for this apparent oversimplification is simply due to the scale of the problem at hand; with the 150 000 structures in the Molecular Modeling Database (MMDB; Madej et al. 2014) available at the time of this writing, the time that it would take to do an in-depth comparison of each and every one of these structures with all of the other structures in MMDB would make the calculations both impractical and intractable.
- Next, the algorithm attempts to optimally align these sets of vectors, looking for pairs of structural elements that are of the same type and relative orientation, with consistent connectivity between the individual elements. The object is to identify highly similar “core substructures,” pairs that represent a statistically significant match above that which would be obtained by comparing randomly chosen proteins with one another.
- Finally, a refinement is done using Monte Carlo (random search) methods at each residue position to optimize the structural alignment. The resultant alignment need not be global, as matches may be between individual structural domains of the proteins being compared.

In 2014, a significant improvement to VAST was introduced. This new approach, called VAST+ (Madej et al. 2014), moves beyond assessing structural similarity by comparing individual three-dimensional domains with one another; instead, it considers the entire set of three-dimensional domains within a macromolecular complex. This approach essentially moves the comparison from the tertiary structure to the quaternary structure level, enabling the identification of similar functional, multi-subunit assemblies. In the VAST+ parlance, macromolecular complexes are referred to as a “biological unit” and can include not just the

proteins that constitute the complex, but also nucleotides and chemicals where such structural information is available. The VAST+ comparison begins as described above for VAST and then marches through a number of steps that involve the identification of biological units that can be superimposed, calculation of root-mean-square deviations (RMSDs) of the superimposed structures as a quantitative measure of the superposition (see Box 12.1), and, finally, performs a refinement step to improve the RMSD values for the superposition. The result of this process is a global structural alignment where both the most and least similar parts of the aligned molecules can be identified and, from a biological standpoint, comparisons between similarly shaped proteins can be facilitated; it can also be used in the context of looking at conformational changes of a single complex under varying conditions. While VAST+ is now the default method for identifying structural neighbors within the Entrez system, keep in mind that the algorithm depends on biological units being explicitly identified within the source Protein Data Bank (PDB) coordinate data records that form the basis for MMDB records; if no such biological units are defined, the original VAST algorithm is then used for the comparisons.

By using approaches such as VAST and VAST+, it is possible to find structural relationships between proteins in cases where simply looking at sequence similarity may not suggest relatedness – information that could, with additional data and insights, be used to help inform the question of functional similarity. More information on additional structure prediction methods based on X-ray or nuclear magnetic resonance (NMR) coordinate data can be found in Chapter 12.

Weighted Key Terms The problem of comparing sequence or structure data somewhat pales next to that of comparing PubMed entries, which consist of free text whose rules of syntax are not necessarily fixed. Given that no two people’s writing styles are exactly the same, finding a way to compare seemingly disparate blocks of text poses a substantial problem. Entrez employs a method known as the relevance pairs model of retrieval to make such comparisons, relying on weighted key terms (Wilbur and Coffee 1994; Wilbur and Yang 1996). This concept is best described by example. Consider two manuscripts with the following titles:

BRCA1 as a Genetic Marker for Breast Cancer

Genetic Factors in the Familial Transmission of the Breast Cancer BRCA1 Gene

Both titles contain the terms *BRCA1*, *Breast*, and *Cancer*, and the presence of these common terms may indicate that the manuscripts are similar in subject matter. The proximity between the words is also considered, so that words common to two records that are closer together are scored higher than common words that are further apart. In the example, the terms *Breast* and *Cancer* are always next to each other, so they would score higher based on proximity than either of those words would against *BRCA1*. Common words found in a title score higher than those found in an abstract, since title words are presumed to be “more important” than those found in the body of an abstract. Overall, weighting depends inversely on the frequency of a given word among all the entries in PubMed, with words that occur infrequently in the database assigned a higher weight while common words are down-weighted.

Hard Links

The hard link concept is simpler and much more straightforward than the neighboring approaches described above. Hard links are applied between entries in *different* databases and exist wherever there is a logical connection between entries. For instance, if a PubMed entry describes the sequencing of a chromosomal region containing a gene of interest, a hard link is established between the PubMed entry and the corresponding nucleotide entry for that gene. If an open reading frame in that gene codes for a known protein, a hard link is established between the nucleotide entry and the protein entry. If the protein entry has an experimentally deduced structure, a hard link would be placed between the protein entry and the structural entry.

Searches can begin anywhere within the Entrez ecosystem – there are no constraints on the user as to where the foray into this information space must begin. However, depending on which database is used as the jumping-off point, different database fields will be available for

searching. This stands to reason, as the entries in different databases are necessarily organized differently, reflecting the biological nature of the entities that each database is trying to catalog.

The Entrez Discovery Pathway

The best way to illustrate the integrated nature of the Entrez system and to drive home the power of neighboring is by considering some biological examples. The simplest way to query Entrez is through the use of individual search terms, coupled together by Boolean operators such as AND, OR, or NOT. Consider the case in which one wants to retrieve all available information on a gene named *DCC* (deleted in colorectal carcinoma), limiting the returned information to publications where an investigator named Guy A. Rouleau is an author. There is a very simple query interface at the top of the NCBI home page, allowing the user to select which database they want to search from a pull-down menu and a text box where the query terms can be entered. In this case, to search for published papers, PubMed would be selected from the pull-down menu and, within the text box to the right, the user would type *DCC AND "Rouleau GA" [AU]*. The [AU] qualifying the second search term indicates to Entrez that this is an *author* term, so only the author field in entries should be considered when evaluating this part of the search statement. The result of the query is shown in Figure 2.2. Here,

The screenshot shows the NCBI PubMed search results page. At the top, the search query is "DCC AND 'Rouleau GA' [AU]". The results are sorted by "Most recent" and show 3 items. The first result is titled "Identification of a homozygous splice site mutation in the dynein axonemal light chain 4 gene on 22q13.1 in a large consanguineous family from Pakistan with congenital mirror movement disorder" by Ahmed I, Mittal K, Sheikh TI, Vasli N, Rafiq MA, Mikhailov A, Ohadi M, Mahmood H, Rouleau GA, Bhatti A, Ayub M, Srour M, John P, Vincent JB. The second result is titled "Partitioning the heritability of Tourette syndrome and obsessive compulsive disorder reveals differences in genetic architecture" by Davis LK, Yu D, Keenan CL, Gamazon ER, Konkashbaev AI, Derks EM, Neale BM, Yang J, Lee SH, Evans P, Barr CL, Bellodi L, Benarroch F, Berrio GB, Bienvenu OJ, Bloch MH, Blom RM, Bruun RD, Budman CL, Camarena B, Campbell D, Cappi C, Cardona Silgado JC, Cath DC, Cavallini MC, Chavira DA, Chouinard S, Conti DV, Cook EH, Coric V, Cullen BA, Deforce D, Delorme R, Dion Y, Edlund CK, Egberts K, Falkai P, Fernandez TV, Gallagher PJ, Garrido H, Geller D, Girard SL, Grabe HJ, Grados MA, Greenberg BD, Gross-Tsur V, Haddad S, Heiman GA, Hemmings SM, Hounie AG, Illmann C, Jankovic J, Jenike MA, Kennedy JL, King RA, Kremeyer B, Kurlan R, Lanzagorta N, Leboyer M, Leckman JF, Lennertz L, Liu C, Lochner C, Lowe TL, Macciardi F, McCracken JT, McGrath LM, Mesa Restrepo SC, Moessner R, Morgan J, Muller H, Murphy DL, Naarden AL, Ochoa WC, Ophoff RA, Osiecki L, Pakstis AJ, Pato MT, Pato CN, Piacentini J, Pittenger C, Pollak Y, Rauch SL, Renner TJ, Reus VI, Richter MA, Riddle MA, Robertson MM, Romero R, Rosário MC, Rosenberg D, Rouleau GA, Ruhmann S, Ruiz-Linares A, Sampaio AS, Samuels J, Sandor P, Sheppard B, Singer HS, Smit JH, Stein DJ, Strengman E, Tischfield JA, Valencia Duarte AV, Vallada H, Van Nieuwerburgh F, Veenstra-Vanderweele J, Walitza S, Wang Y, Wendland JR, Westenberg HG, Shugart YY, Miguel EC, McMahon W, Wagner M, Nicolini H, Posthuma D, Hanna GL, Heutink P, Denys D, Arnold PD, Oostra BA, Nestadt G, Freimer NB, Pauls DL, Wray NR, Stewart SE, Mathews CA, Knowles JA, Cox NJ, Scharf JM. The third result is titled "Mutations in DCC cause congenital mirror movements" by Srour M, Rivière JB, Pham JM, Dubé MP, Girard S, Morin S, Dion PA, Asselin G, Rochefort D, Hince P, Diab S, Sharafaddinzadeh N, Chouinard S, Théoret H, Charron F, Rouleau GA.

Figure 2.2 Results of a text-based Entrez query against PubMed using Boolean operators and field delimiters. The initial query (*DCC AND "Rouleau GA" [AU]*) is shown in the search box near the top of the window, with the three papers identified using this query following below. Each entry gives the title of the manuscript, the names of the authors, and the citation information. The actual record can be retrieved by clicking on the name of the manuscript.

Table 2.1 Entrez Boolean search statements.**General syntax:**

search term **[tag]** Boolean operator search term **[tag]** ...

where **[tag]** =

[ACCN]	Accession
[AD]	Affiliation
[ALL]	All fields
[AU]	Author name
	Lentz R [AU] yields all of Lentz RA, Lentz RB, etc.
	"Lentz R" [AU] yields only Lentz R
[AUID]	Unique author identifier, such as an ORCID ID
[ECNO]	Enzyme Commission numbers
[EDAT]	Entrez date
	YYYY/MM/DD , YYYY/MM, or YYYY; insert a colon for date range, e.g. 2016:2018
[GENE]	Gene name
[ISS]	Issue of journal
[JOUR]	Journal title, official abbreviation, or ISSN number
	Journal of Biological Chemistry J Biol Chem 0021-9258
[LA]	Language
[MAJR]	MeSH major topic
	One of the major topics discussed in the article
[MH]	MeSH terms
	Controlled vocabulary of biomedical terms (subject)
[ORGN]	Organism
[PDAT]	Publication date
	YYYY/MM/DD , YYYY/MM, or YYYY; insert a colon for date range, e.g. 2016:2018
[PMID]	PubMed ID
[PROT]	Protein name (for sequence records)
[PT]	Publication type, includes:
	Review Clinical Trial Lectures Letter Technical Report
[SH]	MeSH subheading
	Used to modify MeSH Terms stenosis [MH] AND pharmacology [SH]
[SUBS]	Substance name
	Name of chemical discussed in article
[SI]	Secondary source ID
	Names of secondary source databanks and/or accession numbers of sequences discussed in article
[TITL]	Title word
	Only words in the definition line (not available in Structure database)
[WORD]	Text words
	All words and numbers in the title and abstract, MeSH terms, subheadings, chemical substance names, personal name as subject, and MEDLINE secondary sources
[VOL]	Volume of journal

and Boolean operator = AND, OR, or NOT

three entries matching the query were found in PubMed. The user can further narrow down the query by adding additional terms if the user is interested in a more specific aspect of this gene or if there are quite simply too many entries returned by the initial query. A list of available field delimiters is given in Table 2.1.

For each of the found papers shown in the Summary view in Figure 2.2, the user is presented with the title of the paper, the authors of that paper, and the citation. To look at any of the papers resulting from the search, the user can simply click on any of the hyperlinked titles. For this example, consider the third reference in the list, by Srour et al. (2010). Clicking on the title takes the user to the Abstract view shown in Figure 2.3. This view presents the name of the paper, the list of authors, their institutional affiliation, and the abstract itself. Below the abstract is a gray bar labeled “MeSH terms, Substances”; clicking on the plus sign at the end of the gray bar reveals cataloging information (MeSH terms, for *medical subject headings*) and indexed substances related to the manuscript. Several alternative formats are available for displaying this information, and these various formats can be selected using the Format pull-down menu found in the upper left corner of the window. Switching to MEDLINE format

The screenshot shows the PubMed website interface. At the top, there is a search bar with 'PubMed' selected and a 'Search' button. Below the search bar, the page is titled 'Format: Abstract -'. The main content area displays the following information:

- Citation:** Science, 2010 Apr 30;328(5978):592. doi: 10.1126/science.1186463.
- Title:** Mutations in DCC cause congenital mirror movements.
- Authors:** Srour M¹, Rivière JB, Pham JM, Dubé MP, Girard S, Morin S, Dion PA, Asselin G, Rochefort D, Hince P, Diab S, Sharafaddinzadeh N, Chouinard S, Théoret H, Charron F, Rouleau GA.
- Author information:** A link to expand this section.
- Abstract:** Mirror movements are involuntary contralateral movements that mirror voluntary ones and are often associated with defects in midline crossing of the developing central nervous system. We studied two large families, one French Canadian and one Iranian, in which isolated congenital mirror movements were inherited as an autosomal dominant trait. We found that affected individuals carried protein-truncating mutations in DCC (deleted in colorectal carcinoma), a gene on chromosome 18q21.2 that encodes a receptor for netrin-1, a diffusible protein that helps guide axon growth across the midline. Functional analysis of the mutant DCC protein from the French Canadian family revealed a defect in netrin-1 binding. Thus, DCC has an important role in lateralization of the human nervous system.
- PMID:** 20431009 **DOI:** 10.1126/science.1186463
- Links:** [Indexed for MEDLINE] **Free full text**
- MeSH terms, Substances:** A list of terms including Axons/physiology, Codon Terminator, DCC Receptor, Dyskinesias/congenital*, Dyskinesias/genetics*, Female, Frameshift Mutation*, Functional Laterality, Genes, DCC*, Genes, Dominant, Genome-Wide Association Study, Haplotypes, Humans, Male, Mutant Proteins/chemistry, Mutant Proteins/metabolism, Nerve Growth Factors/metabolism, Nervous System/growth & development, Netrin-1, Pedigree, Protein Binding, Receptors, Cell Surface/chemistry, Receptors, Cell Surface/genetics, and Receptors, Cell Surface/metabolism*.

On the right side of the page, there is a 'Send to -' section with a 'Full text links' section containing a 'Science' logo. Below that is a 'Save Items' section with an 'Add to Favorites' button. Further down are 'Similar articles' and 'Cited by 42 PubMed Central articles' sections. At the bottom right, there is a 'Related information' section with links for 'Articles frequently viewed together', 'ClinVar', 'Gene', 'Gene (GeneRIF)', and 'Gene (OMIM)'.

Figure 2.3 An example of a PubMed record in Abstract format, as returned through Entrez. This Abstract view is for the third reference shown in Figure 2.2. This view provides connections to related articles, sequence information, and the full-text journal article through the Discovery Column that runs down the right-hand side of the page. See text for details.

produces the MEDLINE layout, with two-letter codes corresponding to the contents of each field going down the left-hand side of the entry (e.g. the author field is again denoted by the code AU). Lists of entries in this format can be saved to the desktop and easily imported into third-party bibliography management programs.

The column on the right-hand side of this window – aptly named the Discovery Column – provides access to the full-text version of the paper and, more importantly, contains many useful links to additional information related to this manuscript. The Similar articles section provides one of the entry points from which the user can take advantage of the neighboring and hard link relationships described earlier and, in the examples that follow, we will return to this page several times to illustrate a selected cross-section of the kinds of information available to the user. To begin this journey, if the user clicks on the *See all* link at the bottom of that section, Entrez will return a list of 104 references related to the original Rouleau paper at the time of this writing; the first six of these papers are shown in Figure 2.4. The first paper in the list is the same Rouleau paper because, by definition, it is most related to itself (the “parent” entry). The order in which the related papers follow is based on statistical

The screenshot shows the PubMed interface with the following details:

- Page Header:** NCBI Resources, How To, Sign in to NCBI, PubMed.gov, US National Library of Medicine, National Institutes of Health.
- Search Bar:** PubMed, Search, Help.
- Format and Sorting:** Format: Summary, Sort by: Link, Per page: 20.
- Section:** Links from PubMed, Items: 1 to 20 of 104.
- Article List:**
 - Mutations in DCC cause congenital mirror movements.** Srour M, Rivière JB, Pham JM, Dubé MP, Girard S, Morin S, Dion PA, Asselin G, Rochefort D, Hince P, Diab S, Sharafaddinzadeh N, Chouinard S, Théoret H, Charron F, Rouleau GA. *Science*. 2010 Apr 30;328(5978):592. doi: 10.1126/science.1186463. PMID: 20431009. **Free Article**. [Similar articles](#)
 - RAD51 haploinsufficiency causes congenital mirror movements in humans.** Depienne C, Bouteiller D, Méneret A, Billot S, Groppa S, Klebe S, Charbonnier-Beaupel F, Corvol JC, Saraiva JP, Brueggemann N, Bhatia K, Cincotta M, Brochard V, Flamand-Roze C, Carpentier W, Meunier S, Marie Y, Gaussem M, Stevanin G, Wehrle R, Vidailhet M, Klein C, Dusart I, Brice A, Roze E. *Am J Hum Genet*. 2012 Feb 10;90(2):301-7. doi: 10.1016/j.ajhg.2011.12.002. Epub 2012 Feb 2. PMID: 22305526. **Free PMC Article**. [Similar articles](#)
 - Mapping netrin receptor binding reveals domains of Unc5 regulating its tyrosine phosphorylation.** Kruger RP, Lee J, Li W, Guan KL. *J Neurosci*. 2004 Dec 1;24(48):10826-34. PMID: 15574733. **Free Article**. [Similar articles](#)
 - A novel DCC mutation and genetic heterogeneity in congenital mirror movements.** Depienne C, Cincotta M, Billot S, Bouteiller D, Groppa S, Brochard V, Flamand-Roze C, Hubsch C, Meunier S, Giovannelli F, Klebe S, Corvol JC, Vidailhet M, Brice A, Roze E. *Neurology*. 2011 Jan 18;76(3):260-4. doi: 10.1212/WNL.0b013e318207b1e0. PMID: 21242494. [Similar articles](#)
 - The netrin 1 receptors Unc5h3 and Dcc are necessary at multiple choice points for the guidance of corticospinal tract axons.** Finger JH, Bronson RT, Harris B, Johnson K, Przyborski SA, Ackerman SL. *J Neurosci*. 2002 Dec 1;22(23):10346-56. PMID: 12451134. **Free Article**. [Similar articles](#)
 - Defining the ligand specificity of the deleted in colorectal cancer (DCC) receptor.** Haddick PC, Tom I, Luis E, Quiñones G, Wraniak BJ, Ramani SR, Stephan JP, Tessier-Lavigne M, Gonzalez LC. *PLoS One*. 2014 Jan 6;9(1):e84823. doi: 10.1371/journal.pone.0084823. eCollection 2014. PMID: 24488166. **Free Article**. [Similar articles](#)
- Right Side:** Filters: Manage Filters, Sort by: Best match, Most recent, Find related data, Database: Select, Find items, Recent Activity, Turn On, Clear, Activity recording is turned off, Turn recording back on.

Figure 2.4 Neighbors to an entry found in PubMed. The original entry from Figure 2.3 (Srour et al. 2010) is at the top of the list, indicating that this is the parent entry. Additional neighbors to each of the papers in this list can be found by clicking the *Similar articles* link found below each entry. See text for details.

Links from PubMed
Showing Current Items.

DCC DCC netrin 1 receptor [*Homo sapiens* (human)]
Gene ID: 1630, updated on 10-Mar-2018

Summary

Official Symbol DCC provided by HGNC
Official Full Name DCC netrin 1 receptor provided by HGNC
Primary source HGNC:HGNC:2701
See related Ensembl:ENSG00000187323 MIM:120470; Vega:OTTHUMG00000132698
Gene type protein coding
RefSeq status REVIEWED
Organism *Homo sapiens*
Lineage Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Mammalia; Eutheria; Euarchontoglires; Primates; Haplorhini; Catarrhini; Hominidae; Homo
Also known as CRC18; CRCR1; MRMV1; HGPPS2; IGDC1; NTN1R1
Summary This gene encodes a netrin 1 receptor. The transmembrane protein is a member of the immunoglobulin superfamily of cell adhesion molecules, and mediates axon guidance of neuronal growth cones towards sources of netrin 1 ligand. The cytoplasmic tail interacts with the tyrosine kinases Src and focal adhesion kinase (FAK, also known as PTK2) to mediate axon attraction. The protein partially localizes to lipid rafts, and induces apoptosis in the absence of ligand. The protein functions as a tumor suppressor, and is frequently mutated or downregulated in colorectal cancer and esophageal carcinoma. [provided by RefSeq, Oct 2009]
Expression Biased expression in testis (RPKM 2.4), brain (RPKM 1.4) and 2 other tissues [See more](#)
Orthologs [mouse](#) [all](#)

Genomic context

Location: 18q21.2 [See DCC in Genome Data Viewer](#) [Map Viewer](#)

Exon count: 32

Annotation release	Status	Assembly	Chr	Location
108	current	GRCh38.p7 (GCF_000001405.33)	18	NC_000018.10 (52340172..53535903)
105	previous assembly	GRCh37.p13 (GCF_000001405.25)	18	NC_000018.9 (49866542..51062273)

Chromosome 18 - NC_000018.10

Genomic map showing the location of DCC on Chromosome 18. The gene is located on the positive strand between coordinates 52340172 and 53535903. Other features shown include LINC01919, LINC01917, RPL29P32, MIR4528, and MIR476P.

Genomic regions, transcripts, and products

Genomic Sequence: [NC_000018.10 Chromosome 18 Reference GRCh38.p7 Primary Assembly](#)

Table of contents

- Summary
- Genomic context
- Genomic regions, transcripts, and products
- Expression
- Bibliography
- Phenotypes
- Variation
- Pathways from BioSystems
- Interactions
- General gene information
 - Markers, Homology, Gene Ontology
- General protein information
- NCBI Reference Sequences (RefSeq)
- Related sequences
- Additional links

Genome Browsers

- Genome Data Viewer
- Map Viewer
- Variation Viewer (GRCh37.p13)
- Variation Viewer (GRCh38)
- 1000 Genomes Browser (GRCh37.p13)
- Ensembl
- UCSC

Related information

- Order cDNA clone
- 3D structures
- BioAssay by Target (List)
- BioAssay by Target (Summary)
- BioAssay, by Gene target
- BioAssays, RNAi Target, Tested
- BioProjects
- BioSystems
- Books

Figure 2.5 The Entrez Gene page for the *DCC* (deleted in colorectal carcinoma) netrin-1 receptor from human. The entry indicates that this is a protein-coding gene at map location 18q21.2, and information on the genomic context of *DCC*, as well as alternative gene names and information on the encoded protein, is provided. An extensive collection of links to other National Center for Biotechnology Information (NCBI) and external databases is also provided. See text for details.

similarity. Thus, the entry closest to the parent is deemed to be the closest in subject matter to the parent. By scanning the titles, the user can easily find related information on other studies, as well as quickly amass a bibliography of relevant references. This is a particularly useful and time-saving function when one is writing grants or papers, as abstracts can easily be scanned and papers of real interest can be identified quickly.

Returning to the Abstract view presented in Figure 2.3, at the bottom of the Discovery Column is a series of hard-link connections to other databases within the Entrez system that can take the user directly to an extensive set of information related to the content of the publication of interest. Here, selecting the Gene link takes the user to Entrez Gene, a feature of Entrez that provides a wealth of information about the gene in question (Figure 2.5). The data are gathered from a variety of sources, including RefSeq. Here, we see that *DCC* is the official symbol of a protein-coding gene for a netrin-1 receptor in humans. The Genomic context section of this page indicates that the *DCC* is a protein-coding gene at map location 18q21.2.

Immediately below, summary information on the genomic region, transcripts, and products of the *DCC* gene are presented graphically, with genomic coordinates provided. Additional content not shown in the figure can be found by scrolling down the Gene page, where the user will find relevant functional information (such as gene expression data), associated phenotypes, information on protein–protein interactions, pathway information, Gene Ontology assignments, and homologies to similar sequences in selected organisms. Shortcut links to these sections can be found in the Table of contents at the top of the Discovery Column. Further down the Discovery Column are extensive lists of links to additional resources provided through NCBI and other sources. One link of note is the *SNP: Gene View* link, taking the user to data derived from dbSNP (Figure 2.6). The information found within dbSNP goes beyond just single-nucleotide polymorphisms (SNPs), including data on short genetic variations such as short insertions and deletions, short tandem repeats, and microsatellites. Here, we will focus on the table shown in Figure 2.6, which is a straightforward way to view information about individual SNPs. Each SNP entry occupies two or more lines of the table, with one line showing the contig reference (the more common allele) and the other showing the SNP (the

Region	Chr. position	mRNA pos	dbSNP rs#	Heterozygosity	Validation	MAF	Allele origin	3D	Clinically Associated	Clinical Significance	Function	dbSNP allele	Protein residue	Codon pos	Amino acid pos	PubMed
	52340791	620	rs779492300	0.000							missense	A	Lys [K]	1	2	
											missense	C	Gln [Q]	1	2	
											contig reference	G	Glu [E]	1	2	
	52340792	621	rs768588905	0.000							missense	C	Ala [A]	2	2	
											contig reference	A	Glu [E]	2	2	
	52340793	622	rs1047445349	N.D.							synonymous	A	Glu [E]	3	2	
											contig reference	G	Glu [E]	3	2	
	52340795	624	rs117282798	0.001		0.0010					missense	G	Ser [S]	2	3	
											contig reference	A	Asn [N]	2	3	
	52340796	625	rs1005837384	N.D.							missense	G	Lys [K]	3	3	
											contig reference	T	Asn [N]	3	3	
	52340800	629	rs1273414468	N.D.							missense	T	Phe [F]	1	5	
											contig reference	C	Leu [L]	1	5	
	52340802	631	rs1220620096	N.D.							synonymous	C	Leu [L]	3	5	
											contig reference	T	Leu [L]	3	5	
	52340804	633	rs1038209589	N.D.							missense	A	Lys [K]	2	6	
											contig reference	G	Arg [R]	2	6	
	52340806	635	rs547090648	0.000		0.0002					missense	C	Arg [R]	1	7	
											contig reference	T	Cys [C]	1	7	
	52340807	636	rs866002143	N.D.							missense	T	Phe [F]	2	7	
											contig reference	G	Cys [C]	2	7	
	52340815	644	rs1282133083	N.D.							missense	A	Ile [I]	1	10	
											contig reference	G	Val [V]	1	10	
	52340816	645	rs192846998	0.000		0.0002					missense	G	Gly [G]	2	10	
											contig reference	T	Val [V]	2	10	
	52340818	648	rs1297820968	N.D.							missense	A	His [H]	2	11	
											contig reference	C	Pro [P]	2	11	
	52340823	652	rs776881751	0.000							synonymous	A	Lys [K]	3	12	
											contig reference	G	Lys [K]	3	12	

Figure 2.6 A section of the Database of Single Nucleotide Polymorphisms (dbSNP) GeneView page providing information on each SNP identified within the human *DCC* gene. See text for details.

less common allele). Consider the first three lines of the table, showing a contig reference G for which there are two documented SNPs, changing the G at that position to either an A or a C. At the protein level, this changes the amino acid at position 2 of the DCC protein from glutamic acid to lysine (for the G-to-A substitution) or to glutamine (for the G-to-C substitution). These rows are colored red since these are “non-synonymous SNPs” – that is, the SNP produces a discrete change at the amino acid level. In contrast, consider the first set of green rows in the table, with the green indicating that this is a “synonymous SNP,” where the codons for the contig reference (G) and the SNP allele (A) ultimately produce the same amino acid (Glu); this is not altogether surprising, with the SNP being in the wobble position of the codon, where there is often redundancy in the genetic code. Additional information on human SNPs can be found in Chapter 15.

Starting again from the Abstract view shown in Figure 2.3, protein sequences from RefSeq that have been linked to this abstract can be found by clicking on the *Protein (RefSeq)* link found in the Related information section on the right-hand side of the page, producing the view shown in Figure 2.7. Note that all but one of the entries is marked as “predicted”; the final

The screenshot displays the NCBI Protein database search results for the query "netrin receptor DCC isoform X4 [Homo sapiens]". The page shows a list of 6 protein entries, each with a checkbox, a title, and an accession number. The first five entries are marked as "PREDICTED" and have red titles, while the sixth is not. Each entry includes links for "BioProject", "Nucleotide", "Taxonomy", "Related Sequences", "GenPept", "Identical Proteins", "FASTA", and "Graphics". The interface includes a search bar, filters, and a "Send to" menu.

Item	Accession	Length	Status
1.	XP_016881059.1	1102 aa	PREDICTED
2.	XP_016881058.1	1427 aa	PREDICTED
3.	XP_016881057.1	1445 aa	PREDICTED
4.	XP_011524146.1	1102 aa	PREDICTED
5.	XP_011524145.1	1443 aa	PREDICTED
6.	NP_005206.2	1447 aa	netrin receptor DCC [Homo sapiens]

Figure 2.7 Entries in the RefSeq protein database corresponding to the original Srour et al. (2010) entry in Figure 2.3. Entries can be accessed and examined by clicking on any of the accession numbers. See text for details.

netrin receptor DCC [Homo sapiens]
 NCBI Reference Sequence: NP_005206.2
[Identical Proteins](#) [FASTA](#) [Graphics](#)

Go to:

LOCUS	NP_005206	1447 aa	linear	PRI 05-FEB-2018
DEFINITION	netrin receptor DCC [Homo sapiens].			
ACCESSION	NP_005206			
VERSION	NP_005206.2			
DBSOURCE	REFSEQ: accession NM_005215.3			
KEYWORDS	RefSeq.			
SOURCE	Homo sapiens (human)			
ORGANISM	Homo sapiens Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Mammalia; Eutheria; Euarchontoglires; Primates; Haplorrhini; Catarrhini; Hominidae; Homo.			
REFERENCE	1 (residues 1 to 1447)			
AUTHORS	Cavaliere S, Stathis A, Fabbri A, Sonzogni A, Ferrone F, Tamborini E, Pelosi G, de Braud F and Platania M.			
TITLE	Uncommon somatic mutations in metastatic NUT midline carcinoma			
JOURNAL	Tumori 103 (Suppl. 1), e5-e8 (2017)			
PUBMED	28967088			
REMARK	GeneRIF: Somatic DCC mutations are associated with metastatic NUT midline carcinoma. Publication Status: Online-Only			
REFERENCE	2 (residues 1 to 1447)			
AUTHORS	Plooster M, Menon S, Winkle CC, Urbina FL, Monkiewicz C, Phend KD, Weinberg RJ and Gupton SL.			
TITLE	TRIM9-dependent ubiquitination of DCC constrains kinase signaling, exocytosis, and axon branching			
JOURNAL	Mol. Biol. Cell 28 (18), 2374-2385 (2017)			
PUBMED	28701345			
REMARK	GeneRIF: Authors demonstrate that tripartite motif protein 9 (TRIM9)-dependent ubiquitination of DCC blocks the interaction with and phosphorylation of FAK.			
REFERENCE	3 (residues 1 to 1447)			
AUTHORS	Marsh AP, Heron D, Edwards TJ, Quartier A, Galea C, Nava C, Rastetter A, Moutard ML, Anderson V, Bitoun P, Bunt J, Faudet A, Gareil C, Gillies G, Gobius I, Guegan J, Heide S, Keren B, Lesne F, Lukic V, Mandelstam SA, McGillivray G, McIlroy A, Meneret A, Mignot C, Morcom LR, Odent S, Paolino A, Pope K, Riant F, Robinson GA, Spencer-Smith M, Srour M, Stephenson SE, Tankard R, Trouillard O, Welniarz Q, Wood A, Brice A, Rouleau G, Attie-Bitach T, Delatycki MB, Mandel JL, Amor DJ, Roze E, Piton A, Bahlo M, Billette de Villemeur T, Sherr EH, Leventer RJ, Richards LJ, Lockhart PJ and Depienne C.			
TITLE	Mutations in DCC cause isolated agenesis of the corpus callosum with incomplete penetrance			
JOURNAL	Nat. Genet. 49 (4), 511-514 (2017)			
PUBMED	28250454			
REMARK	GeneRIF: Identified DCC mutations in four families.			
REFERENCE	4 (residues 1 to 1447)			
AUTHORS	Li W, Lee J, Vikis HG, Lee SH, Liu G, Aurandt J, Shen TL, Fearon ER, Guan JL, Han M, Rao Y, Hong K and Guan KL.			
TITLE	Activation of FAK and Src are receptor-proximal events required for netrin signaling			

Customize view

Analyze this sequence

- Run BLAST
- Identify Conserved Domains
- Highlight Sequence Features
- Find in this Sequence
- Show in Genome Data Viewer

Protein 3D Structure

Structure of DCC FN456 domains

PDB: 5X83

Source: Homo sapiens

Method: X-ray Diffraction

Resolution: 2.997 Å

[See all 9 structures...](#)

Articles about the DCC gene

- Mutations in DCC cause isolated agenesis of the corpus callosum with incomplete penetrance [Nat Genet. 2017]
- DCC Confers Susceptibility to Depression-like Behaviors in Humans and Mice [Biol Psychiatry. 2017]
- Genetic association of deleted in colorectal carcinoma variants with breast cancer [Oncotarget. 2016]

[See all...](#)

Pathways for the DCC gene

- Colorectal cancer
- Axon guidance
- Role of second messengers in netrin-1 signaling

[See all...](#)

Reference sequence information

- RefSeq genomic sequence
See the genomic reference sequence for the DCC gene (NG_013341.2).
- RefSeq mRNA
See reference mRNA sequence for the DCC gene (NM_005215.3).

Figure 2.8 The RefSeq entry for the netrin receptor, the protein product of the human *DCC* gene. The *FASTA* link at the top of the entry provides quick access to the protein sequence in *FASTA* format, while the *Graphics* link provides access to a graphical view of all of the individual elements captured within the entry's feature table (see Figure 2.9). See text for details.

entry in the list has an accession number beginning with NP, indicating that it contains an experimentally determined or verified sequence (see Box 1.2). Clicking on the first line of that entry (number 6) takes the user to the view shown in Figure 2.8, the RefSeq entry for the netrin receptor, the protein product of the *DCC* gene. The feature table – the section of the GenBank entry listing the location and characteristics of each of the documented biological features found within this protein sequence, such as post-translational modifications, recognizable repeat units, secondary structural regions, and clinically relevant variation – is particularly long in this case. This makes it difficult to determine the relative orientation of the features to one another and may lead the user to miss important interactions or relationships between biological features. Fortunately, a viewer that provides a bird's eye view of the elements found within the feature table is available by clicking on the *Graphics* link at the top of the entry, producing the more accessible display shown in Figure 2.9. Zoom controls are provided, and

The screenshot displays the NCBI RefSeq protein entry for netrin receptor DCC [Homo sapiens] (NP_005206.2). The interface includes a search bar, navigation tabs, and a main display area with a protein sequence viewer. The viewer includes a scale from 100 to 1,400 amino acids, a search bar, and several tracks: Protein Features (netrin receptor DCC), region Features (CDD), site Features (CDD), and site Features (phosphorylation). A pop-up box is visible over the phosphorylation site at position 1,267, providing details: site: phosphorylation, Title: Phosphoserine, by MAPK1 (ECO:000250)UniProtKB:Q63155; propagated from UniProtKB/Swiss-Prot (P43146.2), Location: 1,267, Length: 1, Position: 1,268. The right sidebar contains sections for 'Analyze this sequence', 'Protein 3D Structure', 'Articles about the DCC gene', and 'Reference sequence information'.

Figure 2.9 The same RefSeq entry for the netrin receptor shown in Figure 2.8, now rendered in graphical format. The user can learn more about individual elements displayed in this view by simply hovering the cursor over any of the elements in the display; one such example is shown in the pop-up box at the bottom right, for the phosphorylation site at position 1267 of the sequence. Zoom and navigational controls are at the top of the view window, allowing the user to understand this gene within its broader genomic context.

hovering over any of the elements in the display produces a pop-up containing the specific information for that feature from the GenBank entry.

From here, the user can also enter the structural realm by examining the protein structures that are available through the Discovery Column. Clicking on the *See all 9 structures* link takes the user to the view shown in Figure 2.10, listing structural entries related to the netrin receptor. The second entry is for the crystal structure of a fragment of netrin-1 complexed with the DCC receptor (PDB:4URT; Finci et al. 2014), and clicking on the title of that entry takes the user to the structure summary page shown in Figure 2.11. Starting on the right, the Interactions window shows the relationships between the individual elements in this biological unit, here consisting of the netrin-1 protein (circle A), the DCC receptor (circle B), and five different chemical entities (diamonds 1–5). The three-dimensional structure is shown in the left panel,

The screenshot displays the NCBI Structure database search results. The main content area lists 9 protein structures, each with a 3D visualization and a summary of its properties. The entries are:

- Structure of DCC FN456 domains[SIGNALING PROTEIN]**
Taxonomy: Homo sapiens
Proteins: 4 modified: 2018-01-02
MMDB ID: 154304 PDB ID: 5X83
View in iCn3D Similar Structures PubMed Proteins Conserved Domains
- The crystal structure of a fragment of netrin-1 in complex with FN5- FN6 of DCC[PROTEIN BINDING]**
Taxonomy: Homo sapiens
Proteins: 3 Chemicals: 28 modified: 2018-03-12
MMDB ID: 123125 PDB ID: 4URT
View in iCn3D Similar Structures PubMed Proteins Conserved Domains PubChem Compound
- Structure Of The Human Myosin-x Myth4-ferm Cassette Bound To Its Specific Cargo, Dcc[Motor ProteinAPOPTOSIS]**
Taxonomy: Homo sapiens
Proteins: 2 modified: 2014-01-25
MMDB ID: 91519 PDB ID: 3AU4
View in iCn3D Similar Structures PubMed Proteins Conserved Domains
- Solution Structure Of The Sixth Fibronectin Type Iii Domain Of Human Netrin Receptor Dcc[Apoptosis]**
Taxonomy: Homo sapiens
Proteins: 1 modified: 2009-07-14
MMDB ID: 62770 PDB ID: 2EDE
View in iCn3D Similar Structures Proteins Conserved Domains
- Solution Structure Of The Fifth Fibronectin Type Iii Domain Of Human Netrin Receptor Dcc[Apoptosis]**
Taxonomy: Homo sapiens
Proteins: 1 modified: 2009-07-14
MMDB ID: 62769 PDB ID: 2EDD
View in iCn3D Similar Structures Proteins Conserved Domains
- Solution Structure Of The Fourth Fibronectin Type Iii Domain Of Human Netrin Receptor Dcc[Apoptosis]**
Taxonomy: Homo sapiens
Proteins: 1 modified: 2009-07-14
MMDB ID: 62768 PDB ID: 2EDB
View in iCn3D Similar Structures Proteins Conserved Domains
- Solution Structure Of The Third Fibronectin Type Iii Domain Of Human Netrin Receptor Dcc[Apoptosis]**

On the right side of the page, there are several utility sections:

- Filter your results:** All (9), NMR (6), X-ray (3). Manage Filters
- Refine your results - What's this?**
 - Protein Domain Families:** Families (9), Superfamilies (9)
 - Complexes:** Protein-Protein (3), Protein-Chemical (1)
 - Literature:** PubMed (3), PMC (3)
 - Taxonomy:** (9)
- Find related data:** Database: Select, Find Items
- Recent activity:** Turn On, Clear. Activity recording is turned off. Turn recording back on

Figure 2.10 Protein structures associated with the RefSeq entry for the human netrin receptor shown in Figures 2.8 and 2.9. The description of each structure is hyperlinked, allowing the user to access the structure summary page for that entry (see Figure 2.11). Individual links below each entry allow quick access to related structures and proteins, information on conserved domains, and the iCn3D viewer.

and the structure can be further interrogated by clicking on the square with the diagonal arrow in the bottom left of that panel. This action will launch iCn3D (for “I see in three-D”), a web-based viewer that allows the structure to be rotated and rendering options to enhance visualization, and provides a wide variety of additional options; the reader is referred to the iCn3D online documentation for specifics. In the upper right of the 4URT structure summary page is a link to similar structures, as determined by VAST+. Clicking on the VAST+ link produces the output shown in Figure 2.12, here showing the first 10 of 256 structures deemed to have similar biological units to the query (4URT); the table shown here is sorted by RMSD of all aligned residues (in Å), from smallest to largest.

The screenshot shows the NCBI Structure Summary page for PDB entry 4URT. The page is titled "Structure Summary MMDB" and includes a search bar for PDB or MMDB IDs. The main heading is "4URT: The Crystal Structure of a Fragment of Netrin-1 in Complex With Fn5- FN6 of DCC".

Citation: [The crystal structure of netrin-1 in complex with DCC reveals the bifunctionality of netrin-1 as a guidance cue](#)
 Finci LI, Kruger N, Sun X, Zhang J, Chegkazi M, Wu Y, Schenk G, Mertens HD, Svergun DI, Zhang Y, Wang JH, Meijers R
Neuron (2014) 83 p.839-849

Abstract:
 Netrin-1 is a guidance cue that can trigger either attraction or repulsion effects on migrating axons of neurons, depending on the repertoire of receptors available on the growth cone. How a single chemotropic molecule can act in such contradictory ways

Technical Details:
 PDB ID: 4URT [Download](#)
 MMDB ID: 123125 [?](#)
 PDB Deposition Date: 2014/7/2 [?](#)
 Updated in MMDB: 2015/12 [?](#)
 Experimental Method: x-ray diffraction [?](#)
 Resolution: 3.1 Å [?](#)
 Source Organism: Homo sapiens [?](#)
 Similar Structures: VAST+ [?](#)
[Download sequence data](#) [?](#)

Biological Unit: Asymmetric Unit [?](#)

Biological Unit for 4URT: trimeric; determined by author and by software (PISA) [?](#)

The page includes two main visualizations: a "Molecular Graphic" showing a 3D ribbon model of the protein complex in cyan, blue, and magenta, and an "Interactions" diagram showing a network of nodes (A, B, B_0) and edges representing interactions. A legend indicates that circles represent Proteins, squares represent Nucleotides, and diamonds represent Chemicals.

At the bottom, there is a "Download Structure Data" section with a "Download" button, a "Format" dropdown set to "ASN.1 (Cn3D)", and a "Data Set" dropdown set to "Single 3D struct.". A "Download Cn3D" link is also present.

Figure 2.11 The structure summary page for pdb:4URT, the crystal structure of a fragment of netrin-1 complexed with the DCC receptor (Finci et al. 2014). The entry shows header information from the corresponding Molecular Modeling Database (MMDB) entry, a link to the paper reporting this structure, and the methodology used to determine this structure (here, X-ray diffraction with a resolution of 3.1 Å). See text for details.

Medical Databases

Although the focus of many investigators is on sequence-based data, database cataloging and organizing sequence information are not the only kinds of databases useful to the biomedical research community. An excellent example of such a database that is tremendously useful in genomics is called Online Mendelian Inheritance in Man (OMIM), the electronic version of the venerable catalog of human genes and genetic disorders originally founded by Victor McKusick and first published in 1966 (McKusick 1966, 1998; Amberger et al. 2014). OMIM,



Figure 2.12 A list of structures deemed similar to pdb:4URT using VAST+. The table is sorted by the root-mean-square deviation of all aligned residues (in Å), from smallest to largest. Details on each individual structure in the list can be found by clicking on its Protein Data Bank (PDB) ID number.

which is authored and maintained at The Johns Hopkins University School of Medicine, provides concise textual information from the published literature on most human conditions having a genetic basis, as well as pictures illustrating the condition or disorder (where appropriate), full citation information, and links to a number of useful external resources, some of which will be described below. As will become obvious through the following example, a basic knowledge of OMIM should be part of the armamentarium of physician-scientists with an interest in the clinical aspects of genetic disorders.

OMIM has a defined numbering system in which each entry is assigned a unique number – a “MIM number” – that is similar to an accession number, with certain positions within that number indicating information about the genetic disorder itself. The first digit represents the mode of inheritance of the disorder: 1, 2, and 6 stand for autosomal loci or phenotypes, 3 for X-linked loci or phenotype, 4 for Y-linked loci or phenotype, and 5 for mitochondrial loci or phenotypes. An asterisk (*) preceding a MIM number indicates a gene, a hash sign (#) indicates an entry describing a phenotype, a plus sign (+) indicates that the entry describes

The screenshot displays the NCBI OMIM search results for the query 'DCC'. The search bar at the top shows 'OMIM' and a search button. Below the search bar, there are options for 'Limits' and 'Advanced'. The main content area shows two search results:

- 1. **#157600 - MIRROR MOVEMENTS 1: MRMV1**
Cytogenetic locations: 18q21.2
OMIM: 157600
[Gene summaries](#) [Genetic tests](#) [Medical literature](#)
- 2. ***120470 - DELETED IN COLORECTAL CARCINOMA: DCC**
Cytogenetic locations: 18q21.2
OMIM: 120470
[Gene summaries](#) [Genetic tests](#) [Medical literature](#)

On the right side, there is a 'Filter your results' section showing 'All (2)', 'OMIM UniSTS (1)', and 'OMIM dbSNP (1)'. Below this is a 'Find related data' section with a 'Database' dropdown menu and a 'Find items' button. At the bottom, there is a 'Recent activity' section with 'Turn On' and 'Clear' buttons.

The footer contains navigation links, a 'Support Center' link, and a table of resources:

GETTING STARTED	RESOURCES	POPULAR	FEATURED	NCBI INFORMATION
NCBI Education	Chemicals & Bioassays	PubMed	Genetic Testing Registry	About NCBI
NCBI Help Manual	Data & Software	Bookshelf	PubMed Health	Research at NCBI
NCBI Handbook	DNA & RNA	PubMed Central	GenBank	NCBI News & Blog
Training & Tutorials	Domains & Structures	PubMed Health	Reference Sequences	NCBI FTP Site
Submit Data	Genes & Expression	BLAST	Gene Expression Omnibus	NCBI on Facebook
	Genetics & Medicine	Nucleotide	Genome Data Viewer	NCBI on Twitter
	Genomes & Maps	Genome	Human Genome	NCBI on YouTube
	Homology	SNP	Mouse Genome	
	Literature	Gene	Influenza Virus	
	Proteins	Protein	Primer-BLAST	
	Sequence Analysis	PubChem	Sequence Read Archive	
	Taxonomy			
	Variation			

National Center for Biotechnology Information, U.S. National Library of Medicine
8600 Rockville Pike, Bethesda MD, 20894 USA
[Policies and Guidelines](#) | [Contact](#)

Figure 2.13 Online Mendelian Inheritance in Man (OMIM) entries related to the *DCC* gene. The hash sign (#) preceding the first entry indicates that it is an entry describing a phenotype – here, mirror movements. The second entry is preceded by an asterisk (*), indicating that it is a gene entry – here, for the *DCC* gene.

a gene of known sequence and phenotype, and a percent sign (%) describes a confirmed Mendelian phenotype or locus for which the underlying molecular basis is unknown. If no Mendelian basis has been clearly established for a particular entry, no symbol precedes the MIM number.

Here, we will continue the Entrez example from the previous section, following the *OMIM* (cited) link found in the Discovery Column shown in Figure 2.3. An intermediate landing page will then appear listing two entries, one for the *DCC* gene, the other for a phenotype entry describing mirror movements (Figure 2.13). Clicking on the second entry leads the user to the OMIM page for the *DCC* gene shown in Figure 2.14, with the Text section of the entry providing a comprehensive overview of seminal details regarding the identification of the gene, its structure, relevant biochemical features, mapping information, an overview of the gene's function and molecular genetics, and studies involving animal models. For individuals starting work on a new gene or genetic disorder, this expertly curated section of the OMIM entry should be considered “required reading,” as it presents the most important aspects of any given gene, with

The screenshot shows the OMIM website interface. At the top, there is a search bar with the text "Search OMIM..." and a search icon. Below the search bar, the main content area is divided into several sections. On the left, there is a vertical navigation menu with links such as "Table of Contents", "Title", "Gene-Phenotype Relationships", "Text", "Description", "Cloning and Expression", "Gene Structure", "Biochemical Features", "Mapping", "Gene Function", "Molecular Genetics", "Animal Model", "History", "Allelic Variants", "Table View", "References", "Contributors", "Creation Date", and "Edit History".

The main content area displays the following information:

- *120470** **DELETED IN COLORECTAL CARCINOMA; DCC**
- Title**
- Gene-Phenotype Relationships**
- Text**
- Description**: COLORECTAL CANCER-RELATED CHROMOSOME SEQUENCE 18; CRC18 CRCR1
- Alternative titles; symbols**
- HGNC Approved Gene Symbol: DCC**
- Cytogenetic location: 18q21.2** **Genomic coordinates (GRCh38): 18:52,340,171-53,535,902 (from NCBI)**
- Gene-Phenotype Relationships**

Location	Phenotype	Phenotype MIM number	Inheritance	Phenotype mapping key
18q21.2	Colorectal cancer, somatic	114500		3
	Esophageal carcinoma, somatic	133239		3
	Gaze palsy, familial horizontal, with progressive scoliosis, 2	617542	AR	3
	Mirror movements 1 and /or agenesis of the corpus callosum	157600	AD	3

TEXT

Description

The DCC gene encodes a functional receptor for netrin (NTN1; 601614) and mediates axon outgrowth and the steering response (summary by Li et al., 2004).

Cloning and Expression

Vogelstein et al. (1988) found that chromosome 18 sequences were lost frequently in colorectal carcinomas (73%) and in advanced adenomas (47%), but only occasionally in earlier-stage adenomas (11 to 13%). Taken in connection with other findings of changes in chromosome 17, as well as chromosome 5, these findings suggested a model wherein the steps required for malignancy often involve the activation of a dominantly acting oncogene coupled with the loss of several genes that normally suppress tumorigenesis. The critical area in chromosome 18 appeared to reside between 18q21.3 and the telomere.

Fearon et al. (1990) cloned a contiguous stretch of DNA, comprising 370 kb, from the region of 18q suspected to contain the tumor suppressor gene. Potential exons in the 370-kb region were defined by

On the right side of the page, there is a sidebar with "External Links" and "Variation" sections. The "External Links" section includes links to Genome, DNA, Protein, Gene Info, and Clinical Resources. The "Variation" section includes links to 1000 Genome, ClinVar, ExAC, gnomAD, GWAS Catalog, GWAS Central, HGMD, HCVS, NHLBI EVS, and PharmGKB. There are also links to "Animal Models" and "Cellular Pathways".

Figure 2.14 The Online Mendelian Inheritance in Man (OMIM) entry for the *DCC* gene. Each entry in OMIM includes information such as the gene symbol, alternative names for the disease, a description of the disease, a clinical synopsis, and references. See text for details.

links to the original studies cited within the narrative embedded throughout. A particularly useful feature is the list of allelic variants (Figure 2.15); a short description is given after each allelic variant of the clinical or biochemical outcome of that particular mutation. At the time of this writing, there are over 5200 OMIM entries containing at least one allelic variant that either causes or is associated with a discrete phenotype in humans. Note that the allelic variants shown in Figure 2.15 produce significantly different clinical outcomes – two different types of cancer as well as the motor disorder used throughout this example – an interesting case where different mutations in the same gene lead to distinct genetic disorders.

The studies leading to these and similar observations described in a typical entry often provide the foundation for clinical trials aimed at translating this knowledge into new prevention and treatment strategies. NIH's central information source for clinical trials, aptly named ClinicalTrials.gov, contains data on both publicly and privately funded clinical trials

The screenshot shows the OMIM website interface. At the top, there is a search bar labeled "Search OMIM..." and a navigation menu with items like "About", "Statistics", "Downloads", "Contact Us", "MIMmatch", "Donate", and "Help". Below the search bar, the page displays the OMIM entry for the DCC gene (120470). The main content area is titled "ALLELIC VARIANTS (11 Selected Examples):" and includes two tabs: "Table View" and "ClinVar". Three variants are listed:

- .0001 COLORECTAL CANCER, SOMATIC**: Associated with dbSNP:rs387906555, ExAC:rs387906555, and RCV000018603. The description states that Cho et al. (1994) analyzed 60 colorectal cancers and found a somatically acquired point mutation in intron 13, which had features suggestive of an exon.
- .0002 ESOPHAGEAL CARCINOMA, SOMATIC**: Associated with dbSNP:rs121912967 and RCV000018604. The description notes that Miyake et al. (1994) examined 51 cases of primary esophageal carcinoma and found point mutations and loss of the gene.
- .0003 MIRROR MOVEMENTS 1 AND/OR AGENESIS OF THE CORPUS CALLOSUM**: Associated with RCV000018605. The description mentions a large 4-generation French Canadian family with congenital mirror movements (MRMV1; 157600) identified by Srour et al. (2010).

On the right side of the page, there is a sidebar with "External Links" (Genome, DNA, Protein, Gene Info, Clinical Resources) and "Variation" (1000 Genome, ClinVar, ExAC, gnomAD, GWAS Catalog, GWAS Central, HGMD, HGVS, NHLBI EVS, PharmGKB). Below these are "Animal Models" and "Cellular Pathways".

Figure 2.15 An example of a list of allelic variants that can be found through Online Mendelian Inheritance in Man (OMIM). The figure shows three of the four allelic variants for the *DCC* gene. Two of the documented variants lead to cancers of the digestive tract, while two are associated with a movement disorder. The description under each allelic variant provides information specific to that particular mutation.

being conducted worldwide. Figure 2.16 shows the first eight of more than 4600 clinical trials actively recruiting patients with colorectal cancer at the time of this writing, and clicking on the name of a protocol will bring the user to a page providing information on the study, including the principal investigator's name and contact information. Clicking the *On Map* tab at the top of the page produces a clickable map of the world showing how many clinical trials are being conducted in each region or country (Figure 2.17); this view is useful in identifying trials that are geographically close to a potential study subject's home. While we, as scientists, tend to focus on the types of information discussed throughout the rest of this chapter, the clinical trials site is, unarguably, the most important of the sites covered in this chapter, as it provides a means through which patients with a given genetic or metabolic disorder can

The screenshot shows the ClinicalTrials.gov search results page for 'Colorectal Cancer'. The page header includes the NIH logo and navigation links. The search results section shows 4,615 studies found. A table lists the first 8 trials, with columns for Row, Saved, Status, Study Title, Conditions, Interventions, and Locations. The table is filtered to show 1-10 of 4,615 studies, with 10 studies per page. The first trial is 'Study of TAS-102 Plus Radiation Therapy for the Treatment of the Liver in Patients With Hepatic Metastases From Colorectal Cancer', which is currently recruiting. Other trials include 'A Phase 2 Study of Lamivudine in Patients With p53 Mutant Metastatic Colorectal Cancer', 'ASPIrin Intervention for the REDuction of Colorectal Cancer Risk', 'Enhanced Recovery After Surgery Program for Colorectal Cancer: a Multi-center Study (ERASC1)', 'A Phase III Trial Evaluating Fruquintinib Efficacy and Safety in 3+ Line Colorectal Cancer Patients (FRESCO)', 'LEAC-102 for Advanced Colorectal Cancer', 'Signaling Pathways Targeting Colorectal Cancer in Egypt', and 'A Phase I/II Study for the Safety and Efficacy of Panitumumab in Combination With TAS-102 for Patients With Colorectal Cancer'.

Row	Saved	Status	Study Title	Conditions	Interventions	Locations
1	<input type="checkbox"/>	Recruiting	Study of TAS-102 Plus Radiation Therapy for the Treatment of the Liver in Patients With Hepatic Metastases From Colorectal Cancer	• Colorectal Cancer	• Drug: TAS-102 • Radiation: Photon SBRT	• Massachusetts General Hospital Boston, Massachusetts, United States
2	<input type="checkbox"/>	Recruiting	A Phase 2 Study of Lamivudine in Patients With p53 Mutant Metastatic Colorectal Cancer	• Colorectal Cancer Metastatic	• Drug: Lamivudine	• Massachusetts general Hospital Boston, Massachusetts, United States
3	<input type="checkbox"/>	Enrolling by invitation	ASPIrin Intervention for the REDuction of Colorectal Cancer Risk	• Colorectal Cancer	• Drug: Aspirin • Drug: Placebo for Aspirin	• Massachusetts General Hospital Boston, Massachusetts, United States
4	<input type="checkbox"/>	Recruiting	Enhanced Recovery After Surgery Program for Colorectal Cancer: a Multi-center Study (ERASC1)	• Colorectal Cancer	• Procedure: enhanced recovery after surgery	• The First People's Hospital of Changzhou Changzhou, Jiangsu, China • Changzhou Second People's Hospital Affiliated to Nanjing Medical University Changzhou, Jiangsu, China • The First People's Hospital of Lianyungang City Lianyungang, Jiangsu, China • (and 20 more...)
5	<input type="checkbox"/>	Completed	A Phase III Trial Evaluating Fruquintinib Efficacy and Safety in 3+ Line Colorectal Cancer Patients (FRESCO)	• Colorectal Cancer	• Drug: fruquintinib • Drug: placebo	• Hutchison Medi Pharma Investigational Site Hefei, Anhui, China • Hutchison Medi Pharma Investigational Site Beijing, Beijing, China • Hutchison Medi Pharma Investigational Site Guangzhou, Guangdong, China • (and 12 more...)
6	<input type="checkbox"/>	Not yet recruiting	LEAC-102 for Advanced Colorectal Cancer	• Advanced Colorectal Cancer	• Drug: LEAC-102 500mg capsule and FOLFIRI + Bevacizumab/Cetuximab	
7	<input type="checkbox"/>	Not yet recruiting	Signaling Pathways Targeting Colorectal Cancer in Egypt	• Colorectal Cancer	• Genetic: Markers in tissue samples: (TIGAR , TRIM59, PS3, AKT, GSH)	• Assiut University- faculty of medicine - Medical biochemistry department Assiut, Egypt
8	<input type="checkbox"/>	Active, not recruiting	A Phase I/II Study for the Safety and Efficacy of Panitumumab in Combination With TAS-102 for Patients With Colorectal Cancer	• Colorectal Cancer	• Drug: panitumumab, TAS-102	• Nagoya, Aichi, Japan • Kashiwa, Chiba, Japan • Matsuyama, Ehime, Japan • (and 20 more...)

Figure 2.16 The ClinicalTrials.gov page showing all actively recruiting clinical trials relating to colorectal neoplasms. Information on each trial, including the principal investigator of the trial and qualification criteria for participating in the trial, can be found by clicking on the name of the trial.

receive the latest, cutting-edge treatment – treatment that may make a substantial difference to their quality of life.

Organismal Sequence Databases Beyond NCBI

Although it may appear from this discussion that NCBI is the center of the sequence universe, many specialized genomic databases throughout the world serve specific groups in the scientific community. Often, these databases provide additional information not available elsewhere, such as phenotypes, experimental conditions, strain crosses, and map features. These data are of great importance to these communities, not only because they influence experimental design and interpretation of experimental results but also because the kinds of data they contain do not always fit neatly within the confines of the NCBI data model. Development of specialized databases necessarily ensued (and continues), and these databases are intended to be used as an important adjunct to GenBank and similar global databases. It is impossible

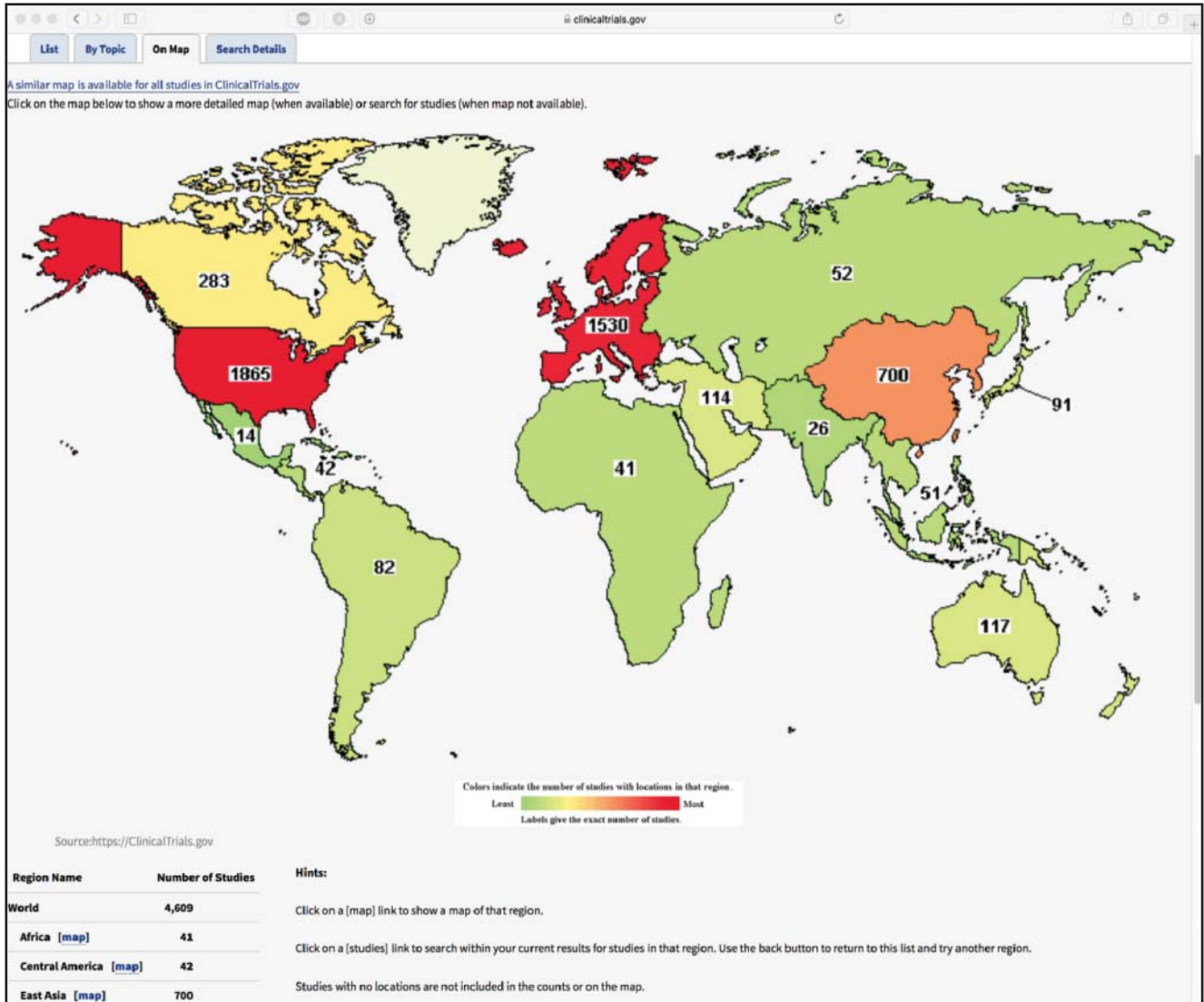


Figure 2.17 A clickable map showing where actively recruiting clinical trials relating to colorectal neoplasms are being conducted. This map-based view of the information presented in Figure 2.15 is useful in identifying trials that are within a reasonable distance of a potential study participant's home.

to discuss the wide variety of such value-added databases here but, to emphasize the sheer number of such databases that exist, the journal *Nucleic Acids Research* devotes its first issue every year to papers describing these databases (Galperin et al. 2017).

An excellent representative example of a specialized organismal database is the Mouse Genome Database (MGD; Bult et al. 2016). Housed at the Jackson Laboratory in Bar Harbor, ME, the MGD provides a curated, comprehensive knowledgebase on the laboratory mouse and is an integral part of its overall Mouse Genome Informatics (MGI) resource. The MGD provides information on genes, genetic markers, mutant alleles and phenotypes, and homologies to other organisms, as well as extensive linkage, cytogenetic, genetic, and physical mapping data. A cross-section of these data is shown in Figure 2.18, providing information on the *Dcc* gene in mouse, the ortholog to the human *DCC* gene from the examples discussed earlier in this chapter. This page can be accessed either by directly searching for the gene name or, in this case, by following links found within the Animal Model section of the OMIM entry for *DCC* discussing seminal discoveries made using

Summary
Symbol *Dcc*
Name deleted in colorectal carcinoma
Synonyms CD30036D22Rik, Igdccl1
Feature Type protein coding gene
IDs MGI:94869
 NCBI Gene: 13176
Gene Overview MyGene.info: DCC
Alliance [gene page](#)

Location & Maps
Sequence Map Chr18:71258738-72351069 bp, - strand
Genetic Map Chromosome 18, 45.24 cM

Homology
Human Ortholog DCC, DCC netrin 1 receptor
Vertebrate Orthologs 9

Human Diseases
Diseases 2 with human DCC associations

Mutations, Alleles, and Phenotypes
Phenotype Summary 27 phenotypes from 5 alleles in 8 genetic backgrounds
 30 phenotypes from multigenic genotypes
 69 phenotype references
All Mutations and Alleles 9
 Gene trapped 1
 Spontaneous 1
 Targeted 6
 Transgenic 1
Incidental Mutations Mutagenetix, APF, CvDC
Find Mice (IMSR) 44 strains or lines available

Gene Ontology (GO) Classifications
All GO Annotations 24
GO References 11
External Resources [FuncBase](#)

Expression
Expression Overview
Other Mouse Links [Allen Institute](#)
[GEO](#)
[Expression Atlas](#)
Other Vertebrate Links [Xenbase dcc](#)
[ZFIN dcc](#)

Figure 2.18 The Mouse Genome Informatics (MGI) entry for the *Dcc* gene in mouse. The entry provides information on the ortholog to the human *DCC* gene, including data on mutant alleles and phenotypes, mapping data, single-nucleotide polymorphisms, and expression data. In the Mutations, Alleles, and Phenotypes section, the phenotype overview uses blue squares to indicate which phenotypes are due to mutations in the *Dcc* gene. In the Expression section, blue squares indicate expression in wild-type mice in the designated tissues, organs, or systems.

mouse models that, in turn, informed our understanding of the effect of *DCC* mutations in humans.

Another long-standing resource devoted to a specific organism is the Zebrafish Model Organism Database, or Zebrafish Information Network (ZFIN) (Howe et al. 2012) – a particularly attractive animal model given the experimental tractability of zebrafish in studying a wide variety of questions focused on vertebrate development, regeneration, inflammation, infectious disease, and drug discovery, to name a few. ZFIN provides a very simple search interface that allows free-text searches using any term. Using *DCC* once again as our search term brings the user to the summary page for the zebrafish *dcc* gene

Gene Name: *DCC netrin 1 receptor*
Gene Symbol: *dcc*
Sequence Ontology ID: SO:0000704
Previous Names: *spaced out, spo, zdcc (1)*
Location: Chr: 5 Mapping Details/Browsers
 Nomenclature History

GENE EXPRESSION
 All Expression Data: 19 figures from 11 publications
 Wild-type Stages, Structures: Segmentation: 1-4 somites (10.33h-11.66h) to Adult (90d-730d, breeding adult)
 brain , CaP motoneuron , cerebellum , diencephalon (all 47) ▶

MUTATIONS AND SEQUENCE TARGETING REAGENTS

Allele	Type	Localization	Consequence	Mutagen	Suppliers
sa12693	Point Mutation	Unknown	Splice Site	ENU	Zebrafish International Resource Center (ZIRC) (order this)
sa13121	Point Mutation	Unknown	Premature Stop	ENU	Zebrafish International Resource Center (ZIRC) (order this)
sa16961	Point Mutation	Unknown	Premature Stop	ENU	Zebrafish International Resource Center (ZIRC) (order this)
sa20326	Point Mutation	Unknown	Splice Site	ENU	Zebrafish International Resource Center (ZIRC) (order this)
sa31409	Point Mutation	Unknown	Premature Stop	ENU	Zebrafish International Resource Center (ZIRC) (order this)

▼ Show all 8 alleles
 Targeting reagents: MO1-dcc (1), MO2-dcc (1), MO3-dcc (1)

PHENOTYPE
 Data: 15 figures from 9 publications
 Observed in: anterior commissure , branching involved in lymph vessel morphogenesis , commissural neuron axon guidance , CoPA axon extension involved in axon guidance (all 27) ▶

DISEASE ASSOCIATED WITH *dcc* HUMAN ORTHOLOG

Disease Ontology Term	OMIM Term	OMIM Phenotype ID
colorectal cancer <input type="checkbox"/>	Colorectal cancer, somatic	114500
esophageal cancer <input type="checkbox"/>	Esophageal carcinoma, somatic	133239
	Mirror movements 1	157600

GENE ONTOLOGY

Ontology	GO Term
Biological Process	anterior/posterior axon guidance <input type="checkbox"/> (more)
Cellular Component	integral component of membrane <input type="checkbox"/> (more)
Molecular Function	netrin receptor activity involved in chemoattraction <input type="checkbox"/> (more)

GO Terms (all 17)

PROTEIN FAMILIES, DOMAINS AND SITES

InterPro:IPR003598 (1)	PROSITE:PS50835 (1)	Pfam:PF00041 (1)
InterPro:IPR003599 (1)	PROSITE:PS50853 (1)	Pfam:PF06583 (1)
InterPro:IPR003961 (1)		Pfam:PF07679 (1)
InterPro:IPR007110 (1)		
InterPro:IPR009138 (1)		
InterPro:IPR010560 (1)		
InterPro:IPR013098 (1)		
InterPro:IPR013783 (1)		
InterPro:IPR021157 (1)		
InterPro:IPR022422 (1)		

Figure 2.19 The Zebrafish Information Network (ZFIN) gene page for the *dcc* gene in zebrafish. This entry provides information on the ortholog to the human *DCC* gene. See text for details.

(Figure 2.19), providing information on zebrafish mutants, sequence targeting reagents, transgenic constructs, orthology to other organisms, data on protein domains found within the Dcc protein product, and annotated gene expression and phenotype data derived from the literature or from direct submissions by members of the zebrafish community. Here, by following the link to the 19 figures in the Gene Expression section, one can examine full-size images illustrating expression patterns for *dcc* under various experimental conditions (Figure 2.20).

While MGD and ZFIN are excellent examples of model organism databases, every major model organism community maintains such a resource. These groups also collaborate to develop central portals to ease information retrieval across many of these resources through the Alliance of Genome Resources.

Figure Gallery (8 Images)

Expression Pattern Search Results for *dcc*
(19 figures with expression from 11 publications)
[Show only figures with images]

Publication (current status)	Data	Fish	Stage Range	Anatomy
Rosenberg <i>et al.</i> , 2014	Fig. 8	mI2Tg <input type="checkbox"/> nkuasgfp1aTg; psi5Tg <input type="checkbox"/>	Day 5	myelinating Schwann cell <input type="checkbox"/> , spinal cord motor neuron <input type="checkbox"/>
Gao <i>et al.</i> , 2012	Fig. 2	WT <input type="checkbox"/>	20-25 somites to Prim-25	dorsal telencephalon <input type="checkbox"/> , dorso-rostral cluster <input type="checkbox"/>
Lakhina <i>et al.</i> , 2012	Fig. 2	p201Tg; p202Tg <input type="checkbox"/>	Prim-5	olfactory placode <input type="checkbox"/> , olfactory receptor cell axon <input type="checkbox"/> , telencephalon medial-lateral axis <input type="checkbox"/>
Zhang <i>et al.</i> , 2012	Fig. 3 Fig. 7	WT <input type="checkbox"/> WT + MO1- <i>dcc</i> <input type="checkbox"/> WT + MO5- <i>robo2</i> <input type="checkbox"/> WT + MO1- <i>ntn1b</i> + MO4- <i>ntn1a</i> <input type="checkbox"/> WT + MO1- <i>sllt2</i> + MO1- <i>sllt3</i> <input type="checkbox"/>	Prim-5 to Prim-25 Prim-25	dorsal telencephalon dorso-medial region <input type="checkbox"/> dorso-rostral cluster <input type="checkbox"/>
Hale <i>et al.</i> , 2011	Fig. 5	WT <input type="checkbox"/>	Prim-5	CaP motoneuron <input type="checkbox"/> , ViP motor neuron <input type="checkbox"/>
Lim <i>et al.</i> , 2011	Fig. 5	WT <input type="checkbox"/> mI2Tg <input type="checkbox"/>	20-25 somites to Prim-15	CaP motoneuron <input type="checkbox"/> , MIP spinal cord neural tube <input type="checkbox"/>
Kastenhuber <i>et al.</i> , 2009	Fig. 3	WT <input type="checkbox"/>	Prim-5	diencephalon dopamine <input type="checkbox"/>
Sull <i>et al.</i> , 2006	text only	WT <input type="checkbox"/>	Prim-25	diencephalon dopamine <input type="checkbox"/>
Fricke <i>et al.</i> , 2005	Fig. 2	rw0Tg <input type="checkbox"/>	Prim-25	hindbrain <input type="checkbox"/>
	Fig. 2	WT <input type="checkbox"/>	20-25 somites to Long-pec	neural tube <input type="checkbox"/> , spinal cord <input type="checkbox"/>
	Fig. 3	WT <input type="checkbox"/>	14-19 somites to Long-pec	hindbrain <input type="checkbox"/>
	Fig. 4	WT <input type="checkbox"/>	Prim-5 to Protruding-mouth	cerebellum <input type="checkbox"/> , optic tectum <input type="checkbox"/>
	Fig. 5	WT <input type="checkbox"/>	Prim-5 to Day 5	gut <input type="checkbox"/> , heart tube <input type="checkbox"/> , me <input type="checkbox"/>
Shen <i>et al.</i> , 2002	text only	WT <input type="checkbox"/>	1-4 somites to Day 5	hindbrain neural plate <input type="checkbox"/>
	Fig. 1	WT <input type="checkbox"/>	5-9 somites to Adult	brain <input type="checkbox"/> , gill <input type="checkbox"/> , heart <input type="checkbox"/>
Hjorth <i>et al.</i> , 2001	Fig. 2	WT <input type="checkbox"/>	5-9 somites to 20-25 somites	epiphysis <input type="checkbox"/> , hindbrain <input type="checkbox"/>
	Fig. 3	WT <input type="checkbox"/>	Prim-5 to Prim-15	diencephalon <input type="checkbox"/> , epiphy <input type="checkbox"/>
	Fig. 4	WT <input type="checkbox"/>	14-19 somites to Prim-5	dorso-rostral cluster <input type="checkbox"/>

A

dcc *dcc*

24hpf lateral 36hpf lateral

B

dcc *dcc/lhx5*

20hpf frontal 20hpf frontal

Figure 2.20 An example of gene expression data available through the Zebrafish Information Network (ZFIN), here showing the expression patterns for the zebrafish *dcc* gene under various experimental conditions. The inset displays a full-size image of data from Gao *et al.* (2012), showing the expression pattern for *dcc* in (panel A) and the co-expression of *dcc* and the Lim homeobox 5 gene (*lhx5*; panel B).

Summary

As alluded to in the introduction to this chapter, the information space available to investigators will continue to expand at breakneck speed, with the size of GenBank alone doubling every year. Although the sheer magnitude of data can present a conundrum to the inexperienced user, mastery of the techniques covered in this chapter will allow researchers in all biological disciplines to make the best use of these data. The movement of modern science to “big data” approaches underscores the idea that both laboratory and computationally based strategies will be necessary in carrying out cutting-edge research. In the same way that investigators are trained in, for example, basic biochemistry and molecular biology methodologies, a basic understanding of bioinformatic techniques as part of the biologist’s arsenal will be indispensable in the future. As is undoubtedly apparent at this point, there is no substitute for placing one’s hands on the computer keyboard to learn how to search and use genomic sequence data effectively. Readers are strongly encouraged to take advantage of the resources

presented here, to grow in confidence and capability by working with the available tools, and to begin to apply bioinformatic methods and strategies toward advancing their own research interests.

Internet Resources

Alliance of Genome Resources	www.alliancegenome.org
Basic Local Alignment Search Tool (BLAST)	ncbi.nlm.nih.gov/BLAST
ClinicalTrials.gov	clinicaltrials.gov
DNA Data Bank of Japan (DDBJ)	www.ddbj.nig.ac.jp
European Molecular Biology Laboratory–European Bioinformatics Institute (EMBL-EBI)	www.ebi.ac.uk
GenBank	www.ncbi.nlm.nih.gov/genbank
iCn3D	www.ncbi.nlm.nih.gov/Structure/icn3d/docs/icn3d_about.html
Mouse Genome Database (MGD)	informatics.jax.org
Online Mendelian Inheritance in Man (OMIM)	omim.org
Protein Data Bank (PDB)	www.rcsb.org/pdb
RefSeq	ncbi.nlm.nih.gov/refseq
Single Nucleotide Polymorphism Database (dbSNP)	www.ncbi.nlm.nih.gov/SNP
Vector Alignment Search Tool (VAST)	www.ncbi.nlm.nih.gov/Structure/VAST
Zebrafish Information Network (ZFIN)	zfin.org

Further Reading

- Baxevanis, A.D. (2012). Searching Online Mendelian Inheritance in Man (OMIM) for information on genetic loci involved in human disease. *Curr. Protoc. Hum. Genet.* Chapter 9, Unit 9.13.1–10. A protocol-driven description of the basic methodology for formulating OMIM searches and a discussion of the types of information available through OMIM, including descriptions of clinical manifestations resulting from genetic abnormalities.
- Galperin, M.Y., Fernández-Suárez, X.M., and Rigden, D.J. (2017). The 24th annual *Nucleic Acids Research* database issue: a look back and upcoming changes. *Nucleic Acids Res.* 45: D1–D11. A curated, annual review of specialized databases of interest and importance to the biomedical research community.

References

- Altschul, S., Gish, W., Miller, W. et al. (1990). Basic local alignment search tool. *J. Mol. Biol.* 215: 403–410.
- Amberger, J.S., Bocchini, C.A., Schiettecatte, F. et al. (2014). OMIM.org: Online Mendelian Inheritance in Man, an online catalog of human genes and genetic disorders. *Nucleic Acids Res.* 43: D789–D798.
- Benson, D.A., Cavanaugh, M., Clark, K. et al. (2017). GenBank. *Nucleic Acids Res.* 45: D37–D42.
- Bult, C.J., Eppig, J.T., Blake, J.A. et al. (2016). Mouse genome database 2016. *Nucleic Acids Res.* 44: D840–D847.
- Collins, F.S., Patrinos, A., Jordan, E. et al., and Members of the DOE and NIH Planning Groups (1998). New goals for the U.S. Human Genome Project: 1998–2003. *Science.* 282: 682–689.
- Collins, F.S., Green, E.D., Guttmacher, A.E., and Guyer, M.S., on behalf of the U.S. National Human Genome Research Institute (2003). A vision for the future of genomics research. *Nature.* 422: 835–847.

- Finci, L.I., Krüger, N., Sun, X. et al. (2014). The crystal structure of netrin-1 in complex with DCC reveals the bifunctionality of netrin-1 as a guidance cue. *Neuron*. 83: 839–849.
- Galperin, M.Y., Fernández-Suárez, X.M., and Rigden, D.J. (2017). The 24th annual *Nucleic Acids Research* database issue: a look back and upcoming changes. *Nucleic Acids Res.* 45: D1–D11.
- Gao, J., Zhang, C., Yang, B. et al. (2012). Dcc regulates asymmetric outgrowth of forebrain neurons in zebrafish. *PLoS One*. 7: e36516.
- Gibrat, J.-F., Madej, T., and Bryant, S. (1996). Surprising similarities in structure comparison. *Curr. Opin. Struct. Biol.* 6: 377–385.
- Green, E.D. and Guyer, M.S., and The National Human Genome Research Institute (2011). Charting a course for genomic medicine from basepairs to bedside. *Nature*. 470: 204–213.
- Howe, D.G., Bradford, Y.M., Conlin, T. et al. (2012). ZFIN, the Zebrafish Model Organism Database: increased support for mutants and transgenics. *Nucleic Acids Res.* 41: D854–D860.
- International Human Genome Sequencing Consortium (2001). Initial sequencing and analysis of the human genome. *Nature*. 409: 860–921.
- Madej, T., Lanczycki, C.J., Zhang, D. et al. (2014). MMDB and VAST+: tracking structural similarities between macromolecular complexes. *Nucleic Acids Res.* 42: D297–D303.
- McKusick, V.A. (1966). *Mendelian Inheritance in Man: Catalogs of Autosomal Dominant, Autosomal Recessive, and X-Linked Phenotypes*. Baltimore, MD: The Johns Hopkins University Press.
- McKusick, V.A. (1998). *Online Mendelian Inheritance in Man: Catalogs of Human Genes and Genetic Disorders*, 12e. Baltimore, MD: The Johns Hopkins University Press.
- Schmutz, J., Wheeler, J., Grimwood, J. et al. (2004). Quality assessment of the human genome sequence. *Nature*. 429: 365–368.
- Srour, M., Rivière, J.B., Pham, J.M.T. et al. (2010). Mutations in DCC cause congenital mirror movements. *Science*. 328: 592.
- Wilbur, W.J. and Coffee, L. (1994). The effectiveness of document neighboring in search enhancement. *Inf. Process. Manag.* 30: 253–266.
- Wilbur, W.J. and Yang, Y. (1996). An analysis of statistical term strength and its use in the indexing and retrieval of molecular biology texts. *Comput. Biol. Med.* 26: 209–222.

This chapter was written by Dr. Andreas D. Baxevanis in his private capacity. No official support or endorsement by the National Institutes of Health or the United States Department of Health and Human Services is intended or should be inferred.

3

Assessing Pairwise Sequence Similarity: BLAST and FASTA

Andreas D. Baxevanis

Introduction

One of the cornerstones of bioinformatics is the process of comparing nucleotide or protein sequences in order to deduce how the sequences are related to one another. Through this type of comparative analysis, one can draw inferences regarding whether two proteins have similar function, contain similar structural motifs, or have a discernible evolutionary relationship. This chapter focuses on *pairwise* alignments, where two sequences are directly compared, position by position, to deduce these relationships. Another approach, *multiple sequence alignment*, is used to identify important features common to three or more sequences; this approach, which is often used to predict secondary structure and functional motifs and to identify conserved positions and residues important to both structure and function, is discussed in Chapter 8.

Before entering into any discussion of how relatedness between nucleotide or protein sequences is assessed, two important terms need to be defined: *similarity* and *homology*. These terms tend to be used interchangeably when, in fact, they mean quite different things and imply quite different biological relationships.

Similarity is a quantitative measure of how related two sequences are to one another. Similarity is always based on an observable – usually pairwise alignment of two sequences. When two sequences are aligned, one can simply count how many residues line up with one another, and this raw count can then be converted to the most commonly used measure of similarity: percent identity. Measures of similarity are used to quantify changes that occur as two sequences diverge over evolutionary time, considering the effect of substitutions, insertions, or deletions. They can also be used to identify residues that are crucial for maintaining a protein's structure or function. In short, a high percentage of sequence similarity may imply a common evolutionary history or a possible commonality in biological function.

In contrast, homology implies an evolutionary relationship and is the putative conclusion reached based on examining the optimal alignment between two sequences and assessing their similarity. Genes (and their protein products) either are or are not homologous – homology is not measured in degrees or percentages. The concept of homology and the term *homolog* may apply to two different types of relationships, as follows.

- If genes are separated by the event of speciation, they are termed *orthologous*. Orthologs are direct descendants of a sequence in a common ancestor, and they may have similar domain structure, three-dimensional structure, and biological function. Put simply, orthologs can be thought of as the same gene (or protein) in different species.
- If genes within the same species are separated by a genetic duplication event, they are termed *paralogous*. The examination of paralogs provides insight into how pre-existing genes may have been adapted or co-opted toward providing a new or modified function within a given species.

The concepts of homology, orthology, and paralogy and methods for determining the evolutionary relationships between sequences are covered in much greater detail in Chapter 9.

Global Versus Local Sequence Alignments

The methods used to assess similarity (and, in turn, infer homology) can be grouped into two types: global sequence alignment and local sequence alignment. Global sequence alignment methods take two sequences and try to come up with the best alignment of the two sequences across their entire length. In general, global sequence alignment methods are most applicable to highly similar sequences of approximately the same length. Although these methods can be applied to any two sequences, as the degree of sequence similarity declines, they will tend to miss important biological relationships between sequences that may not be apparent when considering the sequences in their entirety.

Most biologists instead depend on the second class of alignment algorithm – local sequence alignments. In these methods, the sequence comparison is intended to find the most similar regions within the two sequences being aligned, rather than finding (or forcing) an alignment over the entire length of the two sequences being compared. As such, and by focusing on subsequences of high similarity that are more easily alignable, determining putative biological relationships between the two sequences being compared becomes a much easier proposition. This makes local alignment methods one of the approaches of choice for biological discovery. Often times, these methods will return more than one result for the two sequences being compared, as there may be more than one domain or subsequence common to the sequences being analyzed. Local sequence alignment methods are best for sequences that share some degree of similarity or for sequences of different lengths, and the ensuing discussion will focus mostly on these methods.

Scoring Matrices

Whether one uses a global or local alignment method, once the two sequences under consideration are aligned, how does one actually measure how good the alignment is between “sequence A” and “sequence B”? The first step toward answering that question involves numerical methods that consider not just the position-by-position overlap between two sequences but also the nature and characteristics of the residues or nucleotides being aligned.

Much effort has been devoted to the development of constructs called *scoring matrices*. These matrices are empirical weighting schemes that appear in all analyses involving the comparison of two or more sequences, so it is important to understand how these matrices are constructed and how to choose between matrices. The choice of matrix can (and does) strongly influence the results obtained with most sequence comparison methods.

The most commonly used protein scoring matrices consider the following three major biological factors.

- 1) *Conservation*. The matrices need to consider absolute conservation between protein sequences and also need to provide a way to assess conservative amino acid substitutions. The numbers within the scoring matrix provide a way of representing what amino acid residues are capable of substituting for other residues while not adversely affecting the function of the native protein. From a physicochemical standpoint, characteristics such as residue charge, size, and hydrophobicity (among others) need to be similar.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0	-2	-1	0
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3	-1	0	-1
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3	3	0	-1
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3	4	1	-1
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1	-3	-3	-2
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2	0	3	-1
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2	1	4	-1
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3	-1	-2	-1
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3	0	0	-1
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3	-3	-3	-1
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1	-4	-3	-1
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2	0	1	-1
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1	-3	-1	-1
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1	-3	-3	-1
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2	-2	-1	-2
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2	0	0	0
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-2	-1	1	5	-2	-2	0	-1	-1	-1	0
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3	-4	-3	-2
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1	-3	-2	-1
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4	-3	-2	-1
B	-2	-1	3	4	-3	0	1	-1	0	-3	-4	0	-3	-3	-2	0	-1	-4	-3	-3	4	1	-1
Z	-1	0	0	1	-3	3	4	-2	0	-3	-3	1	-1	-3	-1	0	-1	-3	-2	-2	1	4	-1
X	0	-1	-1	-1	-2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2	0	0	-2	-1	-1	-1	-1	-1

Figure 3.1 The BLOSUM62 scoring matrix (Henikoff and Henikoff 1992). BLOSUM62 is the most widely used scoring matrix for protein analysis and provides best coverage for general-use cases. Standard single-letter codes to the left of each row and at the top of each column specify each of the 20 amino acids. The ambiguity codes B (for asparagine or aspartic acid; Asx) and Z (for glutamine or glutamic acid; Glx) also appear, as well as an X (denoting any amino acid). Note that the matrix is a mirror image of itself with respect to the diagonal. See text for details.

- 2) *Frequency*. In the same way that amino acid residues cannot freely substitute for one another, the matrices also need to reflect how often particular residues occur among the entire constellation of proteins. Residues that are rare are given more weight than residues that are more common.
- 3) *Evolution*. By design, scoring matrices implicitly represent evolutionary patterns, and matrices can be adjusted to favor the detection of closely related or more distantly related proteins. The choice of matrices for different evolutionary distances is discussed below.

There are also subtle nuances that go into constructing a scoring matrix, and these are described in an excellent review by Henikoff and Henikoff (2000).

How these various factors are actually represented within a scoring matrix can be best demonstrated by deconstructing the most commonly used scoring matrix, called BLOSUM62 (Figure 3.1). Each of the 20 amino acids (as well as the standard ambiguity codes) is shown along the top and down the side of a matrix. The scores in the matrix actually represent the logarithm of an odds ratio (Box 3.1) that considers how often a particular residue is observed, in nature, to replace another residue. The odds ratio also considers how often a particular residue would be replaced by another if replacements occurred in a random fashion (purely by chance). Given this, a positive score indicates two residues that are seen to replace each other more often than by chance, and a negative score indicates two residues that are seen to replace each other less frequently than would be expected by chance. Put more simply, frequently observed substitutions have positive scores and infrequently observed substitutions have negative scores.

Box 3.1 Scoring Matrices and the Log Odds Ratio

Protein scoring matrices are derived from the observed replacement frequencies of amino acids for one another. Based on these probabilities, the scoring matrices are generated by applying the following equation:

$$S_{ij} = \log [(q_{ij})/(p_i p_j)]$$

where p_i is the probability with which residue i occurs among all proteins and p_j is the probability with which residue j occurs among all proteins. The quantity q_{ij} represents how often the two amino acids i and j are seen to align with one another in multiple sequence alignments of protein families or in sequences that are known to have a biological relationship. Therefore, the log odds ratio S_{ij} (or “lod score”) represents the ratio of observed vs. random frequency for the substitution of residue i by residue j . For commonly observed substitutions, S_{ij} will be greater than zero. For substitutions that occur less frequently than would be expected by chance, S_{ij} will be less than zero. If the observed frequency and the random frequency are the same, S_{ij} will be zero.

To explain the meaning of the numbers in the matrix more fully, imagine that two sequences have been aligned with one another, and it is now necessary to assess how well a residue in sequence A matches to a residue in sequence B at any given position of the alignment. Using the scoring matrix in Figure 3.1 as our starting point,

- The values on the diagonal represent the score that would be conferred for an exact match at a given position, and these numbers are always positive. So, if a tryptophan residue (W) in sequence A is aligned with a tryptophan residue in sequence B, this match would be conferred 11 points, the value where the row marked **W** intersects the column marked **W**. Also notice that 11 is the highest value on the diagonal, so the high number of points assigned to a W:W alignment reflects not only the exact match but also the fact that tryptophan is the rarest of amino acids found in proteins. Put otherwise, the W:W alignment is much less likely to occur in general and, in turn, is more likely to be correct.
- Moving off the diagonal, consider the case of a conservative substitution: a tyrosine (Y) for a tryptophan. The intersection of the row marked **Y** with the column marked **W** yields a value of 2. The positive value implies that the substitution is observed to occur more often in an alignment than it would by chance, but the replacement is not as good as if the tryptophan residue had been preserved ($2 < 11$) or if the tyrosine residue had been preserved ($2 < 7$).
- Finally, consider the case of a non-conservative substitution: a valine (V) for a tryptophan. The intersection of the row marked **V** with the column marked **W** yields a value of -3 . The negative value implies that the substitution is not observed to occur frequently and may arise more often than not by chance.

Although the meaning of the numbers and relationships within the scoring matrices seems straightforward enough, some value judgments need to be made as to what actually constitutes a conservative or non-conservative substitution and how to assess the frequency of either of those events in nature. This is the major factor that differentiates scoring matrices from one another. To help the reader make an intelligent choice, a discussion of the approach, advantages, and disadvantages of the various available matrices is in order.

PAM Matrices

The first useful matrices for protein sequence analysis were developed by Dayhoff et al. (1978). The basis for these matrices was the examination of substitution patterns in a group of proteins that shared more than 85% sequence identity. The analysis yielded 1572 changes in the 71 groups of closely related proteins that were examined. Using these results, tables were constructed that indicated the frequency of a given amino acid substituting for another amino acid at a given position.

As the sequences examined shared such a high degree of similarity, the resulting frequencies represent what would be expected over short evolutionary distances. Further, given the close evolutionary relationship between these proteins, one would expect that the observed mutations would not significantly change the function of the protein. This is termed *acceptance*: changes that can be accommodated through natural selection and result in a protein with the same or similar function as the original. As individual point mutations were considered, the unit of measure resulting from this analysis is the *point accepted mutation* or PAM unit. One PAM unit corresponds to one amino acid change per 100 residues, or roughly 1% divergence.

Several assumptions went into the construction of the PAM matrices. One of the most important assumptions was that the replacement of an amino acid is independent of previous mutations at the same position. Based on this assumption, the original matrix was extrapolated to come up with predicted substitution frequencies at longer evolutionary distances. For example, the PAM1 matrix could be multiplied by itself 100 times to yield the PAM100 matrix, which would represent what one would expect if there were 100 amino acid changes per 100 residues. (This does not imply that each of the 100 residues has changed, only that there were 100 total changes; some positions could conceivably change and then change back to the original residue.) As the matrices representing longer evolutionary distances are an extrapolation of the original matrix derived from the 1572 observed changes described above, it is important to remember that these matrices are, indeed, predictions and are not based on direct observation. Any errors in the original matrix would be exaggerated in the extrapolated matrices, as the mere act of multiplication would magnify these errors significantly.

There are additional assumptions that the reader should be aware of regarding the construction of these PAM matrices. All sites have been assumed to be equally mutable, replacement has been assumed to be independent of surrounding residues, and there is no consideration of conserved blocks or motifs. The sequences being compared here are of average composition based on the small number of protein sequences available in 1978, so there is a bias toward small, globular proteins, even though efforts have been made to bring in additional sequence data over time (Gonnet et al. 1992; Jones et al. 1992). Finally, there is an implicit assumption that the forces responsible for sequence evolution over shorter time spans are the same as those for longer evolutionary time spans. Although there are significant drawbacks to the PAM matrices, it is important to remember that, given the information available in 1978, the development of these matrices marked an important advance in our ability to quantify the relationships between sequences. As these matrices are still available for use with numerous bioinformatic tools, the reader should keep these potential drawbacks in mind and use them judiciously.

BLOSUM Matrices

In 1992, Steve and Jorja Henikoff took a slightly different approach to the one described above, one that addressed many of the drawbacks of the PAM matrices. The groundwork for the development of new matrices was a study aimed at identifying conserved motifs within families of proteins (Henikoff and Henikoff 1991, 1992). This study led to the creation of the BLOCKS database, which used the concept of a *block* to identify a family of proteins. The idea of a block is derived from the more familiar notion of a motif, which usually refers to a conserved stretch of amino acids that confers a specific function or structure to a protein. When these individual motifs from proteins in the same family can be aligned without introducing a gap, the result is a block, with the term *block* referring to the alignment, not the individual sequences themselves. Obviously, any given protein can contain one or more blocks, corresponding to each of its structural or functional motifs. With these protein blocks in hand, it was then possible to look for substitution patterns only in the most conserved regions of a protein, the regions that (presumably) were least prone to change. Two thousand blocks representing more than 500 groups of related proteins were examined and, based on the substitution patterns in those conserved blocks, *blocks substitution matrices* (or BLOSUMs, for short) were generated.

Given the pace of scientific discovery, many more protein sequences were available in 1992 than in 1978, providing for a more robust base set of data from which to derive these new matrices. However, the most important distinction between the BLOSUM and PAM matrices is that the BLOSUM matrices are directly calculated across varying evolutionary distances and are not extrapolated, providing a more accurate view of substitution patterns (and, in turn, evolutionary forces) at those various distances. The fact that the BLOSUM matrices are calculated directly based only on conserved regions makes these matrices more sensitive to detecting structural or functional substitutions; therefore, the BLOSUM matrices perform demonstrably better than the PAM matrices for local similarity searches (Henikoff and Henikoff 1993).

Returning to the point of directly deriving the various matrices, each BLOSUM matrix is assigned a number (BLOSUM n), and that number represents the conservation level of the sequences that were used to derive that particular matrix. For example, the BLOSUM62 matrix is calculated from sequences sharing no more than 62% identity; sequences with more than 62% identity are clustered and their contribution is weighted to 1. The clustering reduces the contribution of closely related sequences, meaning that there is less bias toward substitutions that occur (and may be over-represented) in the most closely related members of a family. Reducing the value of n yields more distantly related sequences.

Which Matrices Should be Used When?

Although most bioinformatic software will provide users with a default choice of a scoring matrix, the default may not necessarily be the most appropriate choice for the biological question being asked. Table 3.1 is intended to provide some guidance as to the proper selection of scoring matrix, based on studies that have examined the effectiveness of these matrices to detect known biological relationships (Altschul 1991; Henikoff and Henikoff 1993; Wheeler 2003). Note that the numbering schemes for the two matrix families move in opposite directions: more divergent sequences are found using higher numbered PAM matrices and lower numbered BLOSUM matrices. The following equivalencies are useful in relating PAM matrices to BLOSUM matrices (Wheeler 2003):

- PAM250 is equivalent to BLOSUM45
- PAM160 is equivalent to BLOSUM62
- PAM120 is equivalent to BLOSUM80.

In addition to the protein matrices discussed here, there are numerous specialized matrices that are either specific to a particular species, concentrate on particular classes of proteins (e.g. transmembrane proteins), focus on structural substitutions, or use hydrophobicity measures in attempting to assess similarity (see Wheeler 2003). Given this landscape, the most important take-home message for the reader is that no single matrix is the complete answer for all sequence comparisons. A thorough understanding of what each matrix represents is critical to performing proper sequence-based analyses.

Table 3.1 Selecting an appropriate scoring matrix.

Matrix	Best use	Similarity
PAM40	Short alignments that are highly similar	70–90%
PAM160	Detecting members of a protein family	50–60%
PAM250	Longer alignments of more divergent sequences	~30%
BLOSUM90	Short alignments that are highly similar	70–90%
BLOSUM80	Detecting members of a protein family	50–60%
BLOSUM62	Most effective in finding all potential similarities	30–40%
BLOSUM30	Longer alignments of more divergent sequences	<30%

The Similarity column gives the range of similarities that the matrix is able to best detect (Wheeler 2003).

Nucleotide Scoring Matrices

At the nucleotide level, the scoring landscape is much simpler. More often than not, the matrices used here simply count matches and mismatches. These matrices also assume that each of the possible four nucleotide bases occurs with equal frequency (25% of the time). In some cases, ambiguities or chemical similarities between the bases are also considered; this type of matrix is shown in Figure 3.2. The basic differences in the construction of nucleotide and protein scoring matrices should make obvious the fact that protein-based searches are always more powerful than nucleotide-based searches of coding DNA sequences in determining similarity and inferring homology, given the inherently higher information content of the 20-letter amino acid alphabet versus the four-letter nucleotide alphabet.

Gaps and Gap Penalties

Often times, gaps are introduced to improve the alignment between two nucleotide or protein sequences. These gaps compensate for insertions and deletions between the sequences being studied so, in essence, these gaps represent biological events. As such, the number of gaps introduced into a pairwise sequence alignment needs to be kept to a reasonable number so as to not yield a biologically implausible scenario.

The scoring of gaps in pairwise sequence alignments is different from scoring approaches discussed to this point, as no comparison between characters is possible – one sequence has a residue at some position and the other sequence has nothing. The most widely used method for scoring gaps involves a quantity known as the *affine gap penalty*. Here, a fixed deduction is made for introducing the gap; an additional deduction is made that is proportional to the length of the gap. The formula for the affine gap penalty is $G + Ln$, where G is the gap-opening penalty (the cost of creating the gap), L is the gap-extension penalty, and n is the length of the gap, with $G > L$. This last condition is important: given that the gap-opening penalty is larger than the gap-extension penalty, lengthening existing gaps would be favored over creating new ones. The values of G and L can be adjusted manually in most programs to make the insertion

	A	T	G	C	S	W	R	Y	K	M	B	V	H	D	N
A	5	-4	-4	-4	-4	1	1	-4	-4	1	-4	-1	-1	-1	-2
T	-4	5	-4	-4	-4	1	-4	1	1	-4	-1	-4	-1	-1	-2
G	-4	-4	5	-4	1	-4	1	-4	1	-4	-1	-1	-4	-1	-2
C	-4	-4	-4	5	1	-4	-4	1	-4	1	-1	-1	-1	-4	-2
S	-4	-4	1	1	-1	-4	-2	-2	-2	-2	-1	-1	-3	-3	-1
W	1	1	-4	-4	-4	-1	-2	-2	-2	-2	-3	-3	-1	-1	-1
R	1	-4	1	-4	-2	-2	-1	-4	-2	-2	-3	-1	-3	-1	-1
Y	-4	1	-4	1	-2	-2	-4	-1	-2	-2	-1	-3	-1	-3	-1
K	-4	1	1	-4	-2	-2	-2	-2	-1	-4	-1	-3	-3	-1	-1
M	1	-4	-4	1	-2	-2	-2	-2	-4	-1	-3	-1	-1	-3	-1
B	-4	-1	-1	-1	-1	-3	-3	-1	-1	-3	-1	-2	-2	-2	-1
V	-1	-4	-1	-1	-1	-3	-1	-3	-3	-1	-2	-1	-2	-2	-1
H	-1	-1	-4	-1	-3	-1	-3	-1	-3	-1	-2	-2	-1	-2	-1
D	-1	-1	-1	-4	-3	-1	-1	-3	-1	-3	-2	-2	-2	-1	-1
N	-2	-2	-2	-2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

Figure 3.2 A nucleotide scoring table. The scoring for the four nucleotide bases is shown in the upper left of the figure, with the remaining one-letter codes specifying the IUPAC/UBMB codes for ambiguities or chemical similarities. Note that the matrix is a mirror image of itself with respect to the diagonal.

of gaps either more or less permissive, but most methods automatically adjust both G and L to the most appropriate values for the scoring matrix being used.

The other major type of gap penalty used is a *non-affine* (or *linear*) *gap penalty*. Here, there is no cost for opening the gap; a simple, fixed mismatch penalty is assessed for each position in the gap. It is thought that affine penalties better represent the biology underlying the sequence alignments, as affine gap penalties take into account the fact that most conserved regions are ungapped and that a single mutational event could insert or delete many more than just one residue. In practice, use of the affine gap penalty better enables the detection of more distant homologs.

BLAST

By far the most widely used technique for detecting similarity between sequences of interest is the Basic Local Alignment Search Tool, or BLAST (Altschul et al. 1991). The widespread adoption of BLAST as a cornerstone technique in sequence analysis lies in its ability to detect similarities between nucleotide and protein sequences accurately and quickly, without sacrificing sensitivity. The original, standard family of BLAST programs is shown in Table 3.2, but in the time since its introduction many variations of the original BLAST program have been developed to address specific needs in the realm of pairwise sequence comparison, several of which will be discussed in this chapter.

The Algorithm

BLAST is a local alignment method that is capable of detecting not only the best region of local alignment between a query sequence and its target, but also whether there are other plausible alignments between the query and the target. To find these regions of local alignment in a computationally efficient fashion, the method begins by seeding the search with a small subset of letters from the query sequence, known as the *query word*. Using the example shown in Figure 3.3, consider a search where the query word of default length 3 is RDQ. (In practice, all words of length 3 are considered, so, using the sequence in Figure 3.3, the first query word would be TLS, followed by LSH, and so on across the sequence.) BLAST now needs to find not only the word RDQ in all of the sequences in the target database but also related words where conservative substitutions have been introduced, as those matches may also be biologically informative and relevant. To determine which words are related to RDQ, scoring matrices are used to develop what is called the neighborhood. The center panel of Figure 3.3 shows the collection of words that are related to the original query word, in descending score order; the scores here are calculated using a BLOSUM62 scoring matrix (Figure 3.1). Obviously, some cut-off must be applied so that further consideration is only given to words that are indeed closely related to the original query word. The parameter that controls this cut-off is the neighborhood score threshold (T). The value of T is determined automatically by the BLAST program but can be adjusted by the user. Increasing T would push the search toward more exact matches and would speed up the search, but could lead to overlooking possibly interesting biological relationships. Decreasing T allows for the detection of more distant relationships between sequences. Here, only words with $T \geq 11$ move to the next step.

Table 3.2 BLAST algorithms.

Program	Query	Database
BLASTN	Nucleotide	Nucleotide
BLASTP	Protein	Protein
BLASTX	Nucleotide, six-frame translation	Protein
TBLASTN	Protein	Nucleotide, six-frame translation
TBLASTX	Nucleotide, six-frame translation	Nucleotide, six-frame translation

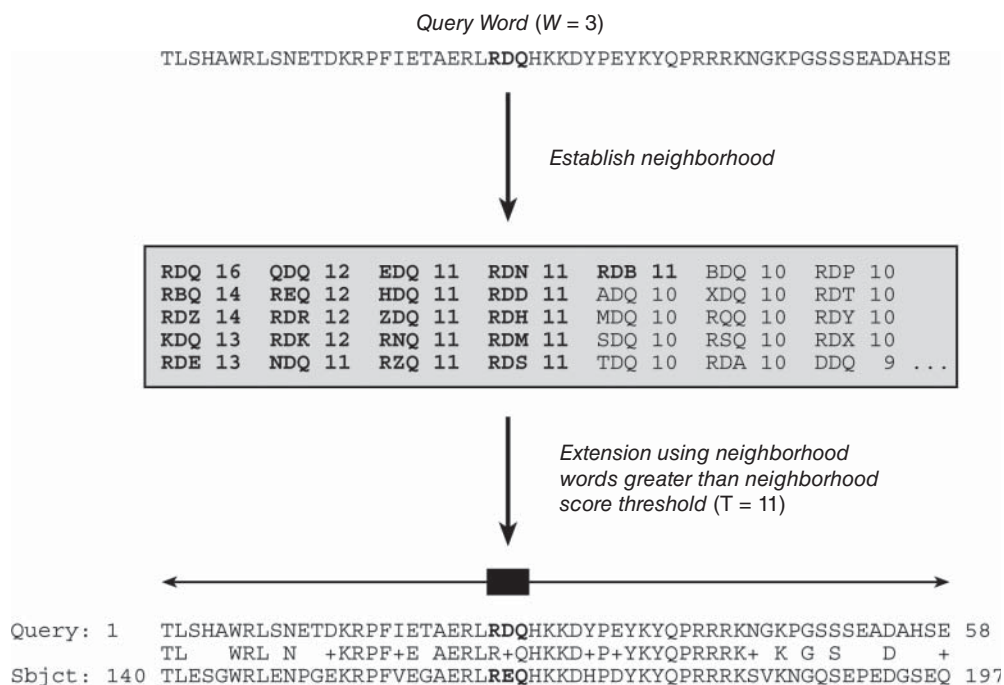


Figure 3.3 The initiation of a BLAST search. The search begins with query words of a given length (here, three amino acids) being compared against a scoring matrix to determine additional three-letter words “in the neighborhood” of the original query word. Any occurrences of these neighborhood words in sequences within the target database are then investigated. See text for details.

Focusing now on the lower panel of Figure 3.3, the original query word (RDQ) has been aligned with another word from the neighborhood whose score is more than the score threshold of $T \geq 11$ (REQ). The BLAST algorithm now attempts to extend this alignment in both directions, tallying a cumulative score resulting from matches, mismatches, and gaps, until it constructs a local alignment of maximal length. Determining what the maximal length actually is can be best explained by considering the graph in Figure 3.4. Here, the number of residues that have been aligned is plotted against the cumulative score resulting from the alignment. The left-most point on the graph represents the alignment of the original query word with one of the words from the neighborhood, again having a value of $T = 11$ or greater. As the extension proceeds, as long as exact matches and conservative substitutions outweigh mismatches

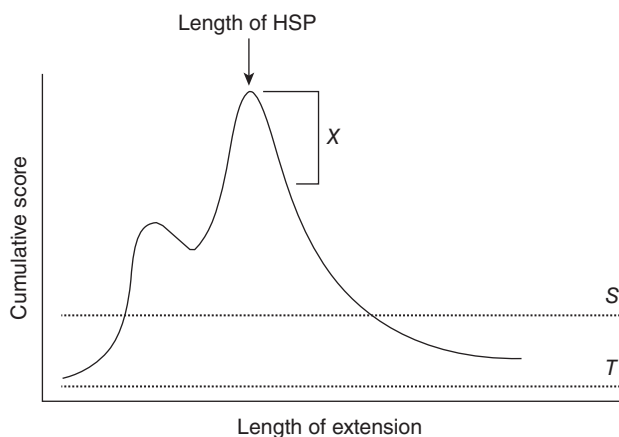


Figure 3.4 BLAST search extension. Length of extension represents the number of characters that have been aligned in a pairwise sequence comparison. Cumulative score represents the sum of the position-by-position scores, as determined by the scoring matrix used for the search. T represents the neighborhood score threshold, S is the minimum score required to return a hit in the BLAST output, and X is the significance decay. See text for details.

and gaps, the cumulative score will increase. As soon as the cumulative score breaks the score threshold S , the alignment is reported in the BLAST output. Simply clearing S does not automatically mean that the alignment is biologically significant, a very important point that will be addressed later in this discussion.

As the extension continues, at some point, mismatches and gaps will begin to outweigh the exact matches and conservative substitutions, accruing negative scores from the scoring matrix. As soon as the curve begins to turn downward, BLAST measures whether the drop-off exceeds a threshold called X . If the curve decays more than is allowed by the value of X , the extension is terminated and the alignment is trimmed back to the length corresponding to the preceding maximum in the curve. The resulting alignment is called a *high-scoring segment pair*, or HSP. Given that the BLAST algorithm systematically marches across the query sequence using all possible query words, it is possible that more than one HSP may be found for any given sequence pair.

After an HSP is identified, it is important to determine whether the resulting alignment is actually significant. Using the cumulative score from the alignment, along with a number of other parameters, a new value called E (for “expect”) is calculated (Box 3.2). For each hit, E gives the number of expected HSPs having a score of S or more that BLAST would find purely by chance. Put another way, the value of E provides a measure of whether the reported HSP is a false positive (see Box 5.4). Lower E values imply greater biological significance.

Box 3.2 The Karlin–Altschul Equation

As one might imagine, assessing the putative biological significance of any given BLAST hit based simply on raw scores is difficult, since the scores are dependent on the composition of the query and target sequences, the length of the sequences, the scoring matrix used to compute the raw scores, and numerous other factors. In one of the most important papers on the theory of local sequence alignment statistics, Karlin and Altschul (1990) presented a formula which directly addresses this problem. The formula, which has come to be known as the Karlin–Altschul equation, uses search-specific parameters to calculate an expectation value (E). This value represents the number of HSPs that would be expected purely by chance. The equation and the parameters used to calculate E are as follows:

$$E = kmNe^{-\lambda S}$$

where k is a minor constant, m is the number of letters in the query, N is the total number of letters in the target database, λ is a constant used to normalize the raw score of the high-scoring segment pair, with the value of λ varying depending on the scoring matrix used; and S is the score of the high-scoring segment pair.

Performing a BLAST Search

While many BLAST servers are available throughout the world, the most widely used portal for these searches is the BLAST home page at the National Center for Biotechnology Information (NCBI; Figure 3.5). The top part of the page provides access to the most frequently performed types of BLAST searches, summarized in Table 3.2, while the lower part of the page is devoted to specialized types of BLAST searches. To illustrate the relative ease with which one can perform a BLAST search, a protein-based search using BLASTP is discussed. Clicking on the *Protein BLAST* box brings users to the BLASTP search page, a portion of which is shown in Figure 3.6. Obviously, a query sequence that will be used as the basis for comparison is required. Harking back to the Entrez discussion in Chapter 2, the sequence of the netrin receptor from *Homo sapiens* (NP_005206.2) has been pasted into the query sequence box. Immediately to the right, the user can use the query subrange boxes to specify whether only a portion of this sequence is to be used; if the whole sequence is to be used, these fields should be left blank.

Basic Local Alignment Search Tool

BLAST finds regions of similarity between biological sequences. The program compares nucleotide or protein sequences to sequence databases and calculates the statistical significance. [Learn more](#)

Web BLAST

Nucleotide BLAST
nucleotide → nucleotide

blastx
translated nucleotide → protein

tblastn
protein → translated nucleotide

Protein BLAST
protein → protein

BLAST Genomes

Enter organism common name, scientific name, or tax id

Human Mouse Rat Microbes

Standalone and API BLAST

Download BLAST
Get BLAST databases and executables

Use BLAST API
Call BLAST from your application

Use BLAST in the cloud
Start an instance at a cloud provider

Specialized searches

SmartBLAST
Find proteins highly similar to your query

Primer-BLAST
Design primers specific to your PCR template

Global Align
Compare two sequences across their entire span (Needleman-Wunsch)

CD-search
Find conserved domains in your sequence

GEO
Find matches to gene expression profiles

IgBLAST
Search immunoglobulins and T cell receptor sequences

VecScreen
Search sequences for vector contamination

CDART
Find sequences with similar conserved domain architecture

Figure 3.5 The National Center for Biotechnology Information (NCBI) BLAST landing page. Examples of the most commonly used queries that can be performed using the BLAST interface are discussed in the text.

Moving to the Choose Search Set section of the page, the database to be searched can be selected using the Database pull-down menu; clicking on the question mark next to the Database pull-down provides a brief description of each of the available target databases. Here, the search will be performed against the RefSeq database (see Box 1.2). Directly below, the Organism box can be used to limit the search results to sequences from individual organisms or taxa. While not part of this worked example, if the user wanted to limit the returned results to those from just mouse and rat, using the same type of syntax used in issuing Entrez searches (see Table 2.1), the user would type `Mus musculus [ORGN] AND Rattus norvegicus [ORGN]` in this field; if the user wanted all results *except* those from mouse and rat, they would also need to check the *Exclude* box. As this search will be performed against RefSeq, one can exclude predicted proteins from the search results by clicking the “Models (XM/XP)” checkbox. Finally, in the Program Selection section, BLASTP is selected by default.

Protein BLAST: search protei: ...

https://blast.ncbi.nlm.nih.gov/Blast.cgi?PROGRAM=blastp&PAGE_TYPE=BlastSearch&LINK_LOC=blasthome

NIH U.S. National Library of Medicine NCBI National Center for Biotechnology Information Sign in to NCBI

BLAST » blastp suite Home Recent Results Saved Strategies Help

Standard Protein BLAST

blastn blastp **blastx** tblastn tblastx

Enter Query Sequence BLASTP programs search protein databases using a protein query. more... Reset page Bookmark

Enter accession number(s), gi(s), or FASTA sequence(s) Clear Query subrange

>NP_005206.2 netrin receptor DCC [Homo sapiens]
 MENSRLRCVWPKLAFVLFGLSFAHLQVTGFQIKAFALRFLSEFSDAVTMRGGNVLLDSCAESDRGVP
 VIKWKKDGIHLALGMDERKQQLSNGSLIQNILHSRHKPDEGLYQCEASLGDSSGSIISRTAKVAVAGPL
 RFLSQTESVTAFMGDTVLLKCEVI GEPMPITIHQKMQDLPVTPGDSRVVVLPSGALQISRLQPGDIGIY
 RCSARNPSSRTGNEAEVRLSDPGLHRQLYFLQRPNSVVAIEGKDAVLECCVSGYPPSPFTWLRGEEVI
 QLRSSKYSLLGGSNLLISNVTDDSGMYTCVVTYKXENISASAELTVLVFPWFNLNHPSNLYAYESMDIEF
 ECTVSGKPVFTYKMKMGDVIIPSDIFQIVGGSNLLILGVVKSDEGPFYQCVAEAEAGNAQTSAGLIVFKP
 AIPSSVLPSPAFRQVVPVFLVSSRPVRLSRPFAEAKGNIQTFVTFPSREGDNRERALNTTPGSLQLTVG

Or, upload file Browse... No file selected.

Job Title Netrin receptor DCC [Homo sapiens]
 Enter a descriptive title for your BLAST search

Align two or more sequences

Choose Search Set

Database Reference proteins (refseq_protein)

Organism Optional
 Enter organism name or id—completions will be suggested Exclude
 Enter organism common name, binomial, or tax id. Only 20 top taxa will be shown.

Exclude Optional
 Models (XM/XP) Uncultured/environmental sample sequences

Entrez Query Optional
 Enter an Entrez query to limit search [YouTube](#) [Create custom database](#)

Program Selection

Algorithm
 blastp (protein-protein BLAST)
 PSI-BLAST (Position-Specific Iterated BLAST)
 PHI-BLAST (Pattern Hit Initiated BLAST)
 DELTA-BLAST (Domain Enhanced Lookup Time Accelerated BLAST)
 Choose a BLAST algorithm

BLAST Search database Reference proteins (refseq_protein) using Blastp (protein-protein BLAST)
 Show results in a new window

Algorithm parameters Note: Parameter values that differ from the default are highlighted in yellow and marked with + sign

Figure 3.6 The upper portion of the BLASTP query page. The first section in the window is used to specify the sequence of interest, whether only a portion of that sequence should be used in performing the search (query subrange), which database should be searched, and which protein-based BLAST algorithm should be used to execute the query. See text for details.

If the user wishes to use the default settings for all algorithm parameters, the search can be submitted by simply clicking on the blue *BLAST* button. However, the user can exert finer control over how the search is performed by changing the items found in the Algorithm parameters section. To access these settings, the user must first click on the plus sign next to the words “Algorithm parameters” to expand this section of the web page, producing the view shown in Figure 3.7. This part of the query page is where the theory underlying a BLAST search discussed earlier in this chapter comes into play. In the General Parameters section, the expect threshold limits returned results to those having an *E* value lower than the specified value, with smaller values providing a more stringent cut-off. The word size setting changes the size of the query word used to initiate the BLAST search, with longer word sizes initiating the search with longer ungapped alignments. A word size of 3 is recommended for protein searches, as shorter words increase sensitivity; however, if searching for near-exact matches, a longer word size can be used, also yielding faster search times.

Protein BLAST: search protei

https://blast.ncbi.nlm.nih.gov/Blast.cgi?PROGRAM=blastp&PAGE_TYPE=BlastSearch&LINK_LOC=blasthome

Program Selection

Algorithm

- blastp (protein-protein BLAST)
- PSI-BLAST (Position-Specific Iterated BLAST)
- PHI-BLAST (Pattern Hit Initiated BLAST)
- DELTA-BLAST (Domain Enhanced Lookup Time Accelerated BLAST)

Choose a BLAST algorithm

BLAST Search database Reference proteins (refseq_protein) using Blastp (protein-protein BLAST)

Show results in a new window

Algorithm parameters Note: Parameter values that differ from the default are highlighted in yellow and marked with ♦ sign

General Parameters

Max target sequences 100 Select the maximum number of aligned sequences to display

Short queries Automatically adjust parameters for short input sequences

Expect threshold ♦ 0.001

Word size ♦ 3

Max matches in a query range 0

Scoring Parameters

Matrix BLOSUM62

Gap Costs Existence: 11 Extension: 1

Compositional adjustments Conditional compositional score matrix adjustment

Filters and Masking

Filter ♦ Low complexity regions

Mask

- Mask for lookup table only
- Mask lower case letters

BLAST Search database Reference proteins (refseq_protein) using Blastp (protein-protein BLAST)

Show results in a new window

Figure 3.7 The lower portion of the BLASTP query page, showing algorithm parameters that the user can adjust to fine-tune the search. Values that have been changed for the search discussed in the text are highlighted in yellow and marked with a diamond. See text for details.

In the Scoring Parameters section, the user can select an appropriate scoring matrix (with the default being BLOSUM62). Changing the matrix automatically changes the gap penalties to values appropriate for that scoring matrix. As described in the discussion of affine gap penalties above, the user may change these values manually; increasing the gap costs would result in pairwise alignments with fewer gaps, where decreasing the values would make the insertion of gaps more permissive.

In the Filters and Masking section, one should filter to remove low-complexity regions. Low-complexity regions are defined simply as regions of biased composition (Wootton and Federhen 1993). These may include homopolymeric runs, short-period repeats, or the subtle over-representation of several residues in a sequence. The biological role of these low-complexity regions is not understood; it is thought that they may represent the results of either DNA replication errors or unequal crossing-over events. It is important to determine whether sequences of interest contain low-complexity regions; they tend to prove problematic when performing sequence alignments and can lead to false-positive results, as they are

generally similar across unrelated proteins. Finally, before issuing the query, be sure to check the box marked “Show results in a new window.” This leaves the original query window (or tab) in place, making it easier to go back and refine or change search parameters, as needed.

Understanding the BLAST Output

The first part of the BLASTP results for the query described above is shown in Figure 3.8. The top part of the figure shows the position of conserved protein domains found by comparing the query sequence with data found within NCBI’s Conserved Domain Database (CDD). This is followed by a graphical overview of the BLASTP results, providing a sense of how many sequences were found to have similarity to the query and how they scored against the query. Details of the various graphical display features are given in the legend to Figure 3.8. The actual list of sequences found as a result of this particular BLASTP search – the “hit list” – is shown, in part, in Figure 3.9. The information included for each hit includes the definition line from



Figure 3.8 Graphical display of BLASTP results. The query sequence is represented by the thick cyan bar labeled “Query,” with the tick marks indicating residue positions within the query. The thinner bars below the query represent each of the matches (“hits”) detected by the BLAST algorithm. The colors represent the relative scores for each hit, with the color key for the scores appearing at the top of the box. The length of each line, as well as its position, represents the region of similarity with the query. Hits connected by a thin line indicate more than one high-scoring segment pair (HSP) within the same sequence; similarly, a thin vertical bar crossing one of the hits indicates a break in the overall alignment. Moving the mouse over any of the lines produces a pop-up that shows the identity of that hit. Clicking on any of the lines takes the user directly to detailed information about that hit (see Figure 3.10).

Sequences producing significant alignments:

Select: All None Selected:0

Alignments Download GenPept Graphics Distance tree of results Multiple alignment

	Description	Max score	Total score	Query cover	E value	Ident	Accession
<input type="checkbox"/>	netrin receptor DCC [Homo sapiens]	2832	2832	100%	0.0	100%	NP_005206.2
<input type="checkbox"/>	netrin receptor DCC precursor [Mus musculus]	2746	2746	100%	0.0	96%	NP_031857.2
<input type="checkbox"/>	netrin receptor DCC precursor [Rattus norvegicus]	2740	2740	100%	0.0	96%	NP_036973.1
<input type="checkbox"/>	deleted in colorectal carcinoma [Xenopus laevis]	2296	2296	100%	0.0	81%	NP_001079254.1
<input type="checkbox"/>	netrin receptor DCC [Danio rerio]	2001	2001	99%	0.0	71%	NP_001030157.1
<input type="checkbox"/>	neogenin isoform 1 precursor [Mus musculus]	1325	1325	97%	0.0	51%	NP_032710.2
<input type="checkbox"/>	neogenin isoform 1 precursor [Homo sapiens]	1321	1321	97%	0.0	51%	NP_002490.2
<input type="checkbox"/>	neogenin isoform 1 precursor [Danio rerio]	1318	1318	98%	0.0	50%	NP_775325.2
<input type="checkbox"/>	neogenin isoform 2 precursor [Homo sapiens]	1308	1308	97%	0.0	51%	NP_001166094.1
<input type="checkbox"/>	neogenin isoform 3 precursor [Homo sapiens]	1304	1304	97%	0.0	51%	NP_001166095.1
<input type="checkbox"/>	neogenin isoform 2 precursor [Danio rerio]	1304	1304	98%	0.0	49%	NP_001315426.1
<input type="checkbox"/>	neogenin isoform 2 precursor [Mus musculus]	1303	1303	97%	0.0	51%	NP_001036217.1
<input type="checkbox"/>	neogenin precursor [Macaca mulatta]	1298	1298	97%	0.0	51%	NP_001248429.1
<input type="checkbox"/>	neogenin [Xenopus tropicalis]	830	889	70%	0.0	52%	NP_001123412.1
<input type="checkbox"/>	neogenin [Saccoglossus kowalevskii]	793	868	94%	0.0	43%	NP_001164709.1
<input type="checkbox"/>	neogenin-like [Aplysia californica]	681	681	71%	0.0	38%	NP_001297427.1
<input type="checkbox"/>	frazzled, isoform B [Drosophila melanogaster]	604	604	71%	0.0	35%	NP_725185.1
<input type="checkbox"/>	Uncharacterized protein CELE_T19B4.7 [Caenorhabditis elegans]	528	528	69%	2e-159	34%	NP_491664.1
<input type="checkbox"/>	frazzled, isoform C [Drosophila melanogaster]	412	684	71%	3e-117	37%	NP_001286353.1
<input type="checkbox"/>	frazzled, isoform A [Drosophila melanogaster]	412	683	71%	3e-117	37%	NP_523716.2
<input type="checkbox"/>	protogenin A precursor [Danio rerio]	268	470	70%	2e-70	29%	NP_001038495.1
<input type="checkbox"/>	protogenin precursor [Rattus norvegicus]	256	445	69%	4e-66	29%	NP_001032740.1
<input type="checkbox"/>	protogenin precursor [Mus musculus]	254	448	69%	1e-65	29%	NP_780694.3
<input type="checkbox"/>	protogenin precursor [Gallus gallus]	247	434	71%	2e-63	28%	NP_001025719.1
<input type="checkbox"/>	protogenin precursor [Homo sapiens]	241	436	69%	2e-61	28%	NP_776175.2

Figure 3.9 The BLASTP “hit list.” For each sequence found, the user is presented with the definition line from the hit’s source database entry, the score value for the best high-scoring segment pair (HSP) alignment, the total of all scores across all HSP alignments, the percentage of the query covered by the HSPs, and the *E* value and percent identity for the best HSP alignment. The hyperlinked accession number allows for direct access to the source database record for that hit. In the *E* value column, vanishingly low *E* values are rounded down to zero. For non-zero *E* values, exponential notation is used; using the first non-zero value in the figure, 2e-159 should be read as 2×10^{-159} .

the hit’s source database entry, the score value that is, in turn, used to calculate the *E* value for the best HSP alignment, the percent identity for that best HSP alignment, and the hyperlinked accession number, allowing for direct access to the source database record for that hit. The table is sorted by *E* value from lowest to highest, by default; recall that lower values of *E* represent better alignments. In the *E* value column, notice that many of the entries have *E*-values of 0.0. This represents a vanishingly low *E* value that has been rounded down to zero and implies statistical significance. Note that each entry in the hit list is preceded by a check box; checking one or more of these boxes lights up the grayed-out options shown in Figure 3.9, allowing the user to download the selected sequences, view the selected hits graphically, generate a dendrogram, or construct a multiple sequence alignment on the fly.

Clicking on the name of any of the proteins in the hit list moves the user down the page to the portion of the output showing the pairwise alignment(s) for that hit (Figure 3.10). The

neuronal cell adhesion molecule precursor [Gallus gallus]
Sequence ID: [NP_990597.1](#) Length: 1268 Number of Matches: 2

Range 1: 254 to 1040 [GenPept](#) [Graphics](#) ▼ Next Match ▲ Previous Match

Score	Expect	Method	Identities	Positives	Gaps
223 bits(568)	1e-55	Compositional matrix adjust.	233/825(28%)	349/825(42%)	73/825(8%)
Query 47	SDAVTMRGGVLLDCAESDRGVPVVKWKDGIHLALGMDERKQQLSNGSLLIQNILHSR				106
Sbjct 254	S+ V +RG +LL+C A + PVI+W K+G L + N ++ I S SNKVELRGNVLLLECIA-AGLPTFVIRWIKEGGELP-----ANRTFFENFKTKLII DVS-				307
Query 107	HHKPEGLYQCEA--SLGDSGSIISRTAKVAVAGPLRFLSQTESVTFMGDTVLLKCEVI				164
Sbjct 308	+ D G Y+C A +LG + +IS T K A + + + + + G+ L C --EADSGNYKCTARNTLGS THHVISVTKAAP----YWITAPRNLVLS PGEDGTLCRAN				361
Query 165	GEPMTIHWKQKQDLTPIPGD-SRVVLPSPGALQISRLQPGDIGIYRCSARNPASSRTG				223
Sbjct 362	G P P+I W N + P D SR V + S +Q +Y+C+A N GNPKPSISWLTNGVPIAIAPEPDSRKV--DGDITIFSAVQERSAVYQCNASNEYG YLLA				419
Query 224	NEAEVRILSDPGLHRQLYFLQRPNSVVAIEGKDAVLECCVSGYPPSPFTWLRGEEVQLR				283
Sbjct 420	N A V +L++P R L + V+A A+++ G P P W R G + L R N-AFVNVLAEP--PRILTANKLYQVIA--DSPALIDCAYFGSPKPEIWFGRGVKGSILR				474
Query 284	SKKYslggenllianVTDDSGMYTCVVYTKNENISASAELTVLPVFPWFLNHPNSLYAY				343
Sbjct 475	+ Y + D +G YTCV K +L V P + P GNEYVFDHNGTLEIPVAQKDSGTGYTCVARNKLGKTKQNEVQLVKDPTMIIKQPKYKVIQ				534
Query 344	ESMDIEFECTVSGKFP--VPTVNMKNGDVVPSDYFQIVGGSNLRILGVKSDGFGYQCV				401
Sbjct 535	S FEC + P +PTV W+K+ + +P D +VG NL I+ V D+G Y C+ RSAQASFECEVIKHDP TLIPTVIWLKNN-ELPDDERFLVGKDNLTIMNVTDKDDGTTYCI				593
Query 402	AENEAGNAQTSACLIV----PKPAIPSSSVLPAPRDVVPVLSRFRVLSWRPPEAKG				457
Sbjct 594	+ SA L V P PAI + P+ P D+ R + LSW P E VNTLDSVSASAVLTVVAAPPTPAIYAR--PNPDLDEL TGLQLERSIELSWVPGENNS				651
Query 458	NIQFTVFFSREGDNRERAL---NTQPGLQLTVGNLKPPEAMYTFRVVAYNEWGPGESE				514
Sbjct 652	I F + + E E + T PGS L P Y+FRV+A NE G S PITNFVIEY--EDGLHEPGVWHYQTEVPGSHTTVQLKLSFYVNSFRVIAVNEIG---RS				706
Query 515	QPIKVAQPELVQVGPVE---NLQAVSTSPSITLITWEPPA--YANGPVQGYRLCTEVS				569
Sbjct 707	QP + + Q + P E N+Q + + P +++ITWE +NGP Y++ + QPSEPEQYLTKSANPENPSNVQIGSEPDNLVITWESLKGQFQSNPGQLQYKVSWRQKD				766
Query 570	TGKE-QNIEVDGLS-YKLEGLKFEYSLRFLAYNRYGPGVSTDDITVVTLSDVPSAPPQ				627
Sbjct 767	E + + V +S Y + G F Y ++ A N G ++ + D+P P VDEWTSVVVANVSKYIVSGTFTFVPEIKVQALNDLGYAPEPSEVIGHSGEDLPMVAPG				826
Query 628	NVSLVNVNSRIKVSWLPSPSGTQNGFITGYKIRH-----RKTTRRGEME--TLEPNN				678
Sbjct 827	NV + V+NS KV W P P + G +GYK+ + R++ R E + T N NVQVHVINSTLAKVHWDPVPLKSVRGLHQYKVVYKWSLRSRKRHVEKILITFRGNK				886
Query 679	LWYFTGLEKGSQYSFQVSAMTVNGTGPSSNYTAETPENDLDESQVDPSSLHV-RPQ				737
Sbjct 887	+ P GLE S Y V + G GP S +TPE +VP PS L + P TFGMLPGLEPYSSYKLNVRVNGKGECPASPKVFKTPEG-----VSPSPFLKITNPT				940
Query 738	TNCIMSWTTPPLNPNivrgy-----iggyvgvSPYAETVRVDSQRYYISIERLESSH				791
Sbjct 941	+ + + W P +PN V+ Y I P E +R+ + + ++ L S+ LDSLTLWEGSPTHPNGVLTSYILKFPQINNTHELGLPVE-IRIPANESSLILKLNLYSTR				999
Query 792	YVISLKAFFNAGEGVPLYESATTRSITDPTDVPDYPLDDFPPTS 836				
Sbjct 1000	Y KFYFNAQTSVSGSGSQITEEAVT--IMDEVQPL--YPRIRNVTTA 1040				

Range 2: 48 to 415 [GenPept](#) [Graphics](#) ▼ Next Match ▲ Previous Match ▲ First Match

Score	Expect	Method	Identities	Positives	Gaps
119 bits(297)	2e-23	Compositional matrix adjust.	101/374(27%)	161/374(43%)	18/374(4%)
Query 46	PSDAVTRMRGGVLLDCAESDRGVPVVKWKDGIHLALGMDERKQQLSNGSLLIQNILHS				105
Sbjct 48	P D + N+++ C A+ + P W ++G H + D + N L+ NI++ PKDVIDVDPRENIIVIQCEAKG-KPPPSFSWTRNGTHFDIDKDAQVTMKPNSGTLVNVIMNG				106
Query 106	RHHKPEGLYQCEASLGDGSIISRTAKVAVA-GPLRFLSQTESVTFMGDTVLLKCE-V				163
Sbjct 107	+ EG+YQC A + G+ IS + + PL + E GD+++L C VKAEAYEGVYQCTAR-NERGAALSNINIVIRPSRSLWTKKLEPNHVRREGDSVLNCRPP				165
Query 164	IGEPMTIHWKQKQDLTPIPGDSRVVLPSPGALQISRLQPGDIGI-YRCSAR--NPASS				220
Sbjct 166	+G P P I W N +P RV +G L S +QF D + Y C AR + + VGLPPPIIFWMDNA--FQRLPQSERVSGQLNGDLYFSNVQPEDTRVDYICARFNHTQTI				223
Query 221	RTGNEAEVRILSDPGLHRQLYFLQRP----SNVVAIEGKDAVLECCVSGYPPSPFTWLR-				275
Sbjct 224	+ V++ S + + L P SN V + G +LEC +G P P W++ QQKQPSIVKVFSTKPVTERPPVLLTPMGSTSNKVELRGNVLLLECIAAGLPTFVIRWIK				283
Query 276	GEEVIQLRSKKYslggenllianVTDDSGMYTCVVYTKNENISASAELTVLPVFPWFLN				335
Sbjct 284	G E+ R+ + L I +V++ DSG Y C + +TV P+++ GGELPANRT--FFENFKTKLII DVS EADSGNYKCTARNTLGS THHVISVTKAAPYWIT				341
Query 336	HPSNLYAYESMDIEFECTVSGKFPVPTVNMKNG--DVPVPSDYFQIVGGSNLRILGVK				393
Sbjct 342	P NL D C +G P P+++W+ NG + P D + V G + V + APRNLVLS PGEDGTLCRANGNPKPSISWLTNGVPIAIAPEPDSRKVDGDTIIFSAVQER				401
Query 394	DEGFYQCVAENAG 407				
Sbjct 402	YQC A NE G SSAVYQCNASNEYG 415				

Figure 3.10 Detailed information on a representative BLASTP hit. The header provides the identity of the hit, as well as the score and *E* value. The percent identity indicates exact matches, whereas the percent “positives” considers both exact matches and conservative substitutions. The gap figures show how many residues remain unaligned owing to the introduction of gaps. Gaps are indicated by dashes and low-complexity regions are indicated by grayed-out lower case letters. Note that there is no header preceding the second alignment; this indicates that this is a second high-scoring segment pair (HSP) within the same database entry.

header provides the complete definition line for this particular hit, and each identified HSP is then shown below the header. In most cases, the user will only see one alignment, but in the case shown in Figure 3.10 there are two, with the hit having the better score and E value shown first. The statistics given for each hit include the E value, the number of identities (exact matches), the number of “positives” (exact matches and conservative substitutions), and the number of residues that fell into a gapped region. Within the alignments, gaps are indicated by dashes, while low-complexity regions are indicated by grayed-out lower case letters.

Suggested BLAST Cut-Offs

As was previously alluded to, the listing of a hit in a BLAST report does not automatically mean that the hit is biologically significant. Over time, and based on both the methodical testing and the personal experience of many investigators, many guidelines have been put forward as being appropriate for establishing a boundary that separates meaningful hits from the rest. For nucleotide-based searches, one should look for E values of 10^{-6} or less and sequence identities of 70% or more. For protein-based searches, one should look for hits with E values of 10^{-3} or less and sequence identities of 25% or more. Using less-stringent cut-offs risks entry into what is called the “twilight zone,” the low-identity region where any conclusions regarding the relationship between two sequences may be questionable at best (Doolittle 1981, 1989; Vogt et al. 1995; Rost 1999).

The reader is cautioned not to use these cut-offs (or any other set of suggested cut-offs) blindly, particularly in the region right around the dividing line. Users should always keep in mind whether the correct scoring matrix was used. Likewise, they should manually inspect the pairwise alignments and investigate the biology behind any putative homology by reading the literature to convince themselves whether hits on either side of the suggested cut-offs actually make good biological sense.

BLAST 2 Sequences

A variation of BLAST called BLAST 2 Sequences can be used to find local alignments between any two protein or nucleotide sequences of interest (Tatusova and Madden 1999). Although the BLAST engine is used to find the best local alignment between the two sequences, no database search is performed. Rather, the two sequences to be compared are specified in advance by the user. The method is particularly useful for comparing sequences that have been determined to be homologous through experimental methods or for making comparisons between sequences from different species. Returning to the Protein BLAST (BLASTP) search page shown in Figure 3.6, checking the box marked “Align two or more sequences” will change the structure of the page, now allowing for the user to enter both the query and subject sequences that will be compared with one another (Figure 3.11). As with any BLAST search, the user can adjust the standard array of BLAST-related options, including the selection of scoring matrix and gap penalties. A sample of the results produced by the BLAST 2 Sequences method is shown in Figure 3.12, comparing the transcription factor SOX-1 from *H. sapiens* and the ctenophore *Mnemiopsis leidyi*, the earliest branching animal species dating back at least 500 million years in evolutionary time (Ryan et al. 2013; Schnitzler et al. 2014). The major difference between this output and the typical BLAST output is the inclusion of a dot matrix view of the alignment, or “dotplot.” Dotplots are intended to provide a graphical representation of the degree of similarity between the two sequences being compared, allowing for the quick identification of regions of local alignment, direct or inverted repeats, insertions, deletions, and low-complexity regions. The dotplot in Figure 3.12 indicates two regions of alignment, and additional information on those two regions of alignment is provided in the Alignments section at the bottom of the figure. As with all BLAST searches, the Alignments section provides the user with the usual set of scores, the E value, and percentages for identities, positives, and any gaps that may have been introduced.

The screenshot displays the NCBI BLAST web interface for a BLAST 2 Sequences alignment. The page title is "Align Sequences Protein BLAST". The "Enter Query Sequence" section is expanded to show a "Enter Subject Sequence" section. The query sequence is for human SOX-1 and the subject sequence is for *Mnemiopsis leidyi*. The "Program Selection" section shows "blastp (protein-protein BLAST)" selected. The "Algorithm parameters" section is expanded to show "General Parameters" with "Max target sequences" set to 100, "Short queries" checked, and "Expect threshold" set to .001.

Figure 3.11 Performing a BLAST 2 Sequences alignment. Clicking the check box at the bottom of the Enter Query Sequence section expands the search page, generating a new Enter Subject Sequence section. Here, sequences for the transcription factor SOX-1 from human and the ctenophore *Mnemiopsis leidyi* have been used as the query and subject, respectively (Schnitzler et al. 2014). Here, only the BLASTP algorithm is available in the Program Selection section, as a one-to-one alignment has already been specified. The usual set of algorithm parameters is available, allowing the user to fine-tune the alignment as needed.

MegaBLAST

MegaBLAST is a variation of the BLASTN algorithm that has been optimized specifically for use in aligning either long or highly similar (>95%) nucleotide sequences and is a method of choice when looking for exact matches in nucleotide databases. The use of a greedy gapped alignment routine (Zhang et al. 2000) allows MegaBLAST to handle longer nucleotide sequences approximately 10 times faster than BLASTN would. MegaBLAST is particularly well suited to finding whether a sequence is part of a larger contig, detecting potential sequencing errors, and for comparing large, similar datasets against each other. The run speeds that are achieved using MegaBLAST come from changing two aspects of the traditional



Figure 3.12 Typical output from a BLAST 2 Sequences alignment, based on the query issued in Figure 3.11. The standard graphical view is shown at the top of the figure, here indicating two high-scoring segment pairs (HSPs) for the alignment of the sequences for the transcription factor SOX-1 from human and the ctenophore *Mnemiopsis leidyi*. The dot matrix view is an alternative view of the alignment, with the query sequence represented on the horizontal axis and the subject sequence represented by the vertical axis; the diagonal indicates the regions of alignment captured within the two HSPs. The detailed alignments are shown at the bottom of the figure, along with the *E* values and alignment statistics for each HSP.

BLASTN routine. First, longer default word lengths are used; in BLASTN, the default word length is 11, whereas MegaBLAST uses a default word length of 28. Second, MegaBLAST uses a non-affine gap penalty scheme, meaning that there is no penalty for opening the gap; there is only a penalty for extending the gap, with a constant charge for each position in the gap. MegaBLAST is capable of accepting batch queries by simply pasting multiple sequences in FASTA format or a list of accession numbers into the query window.

There is also a variation of MegaBLAST called discontinuous MegaBLAST. This version has been designed for comparing divergent sequences from different organisms, sequences where one would expect there to be low sequence identity. This method uses a discontinuous word approach that is quite different from those used by the rest of the programs in the

BLAST suite. Here, rather than looking for query words of a certain length to seed the search, non-consecutive positions are examined over longer sequence segments (Ma et al. 2002). The approach has been shown to find statistically significant alignments even when the degree of similarity between sequences is very low.

PSI-BLAST

The variation of the BLAST algorithm known as PSI-BLAST (for position-specific iterated BLAST) is particularly well suited for identifying distantly related proteins – proteins that may not have been found using the traditional BLASTP method (Altschul et al. 1997; Altschul and Koonin 1998). PSI-BLAST relies on the use of position-specific scoring matrices (PSSMs), which are also often called hidden Markov models or profiles (Schneider et al. 1986; Gribskov et al. 1987; Staden 1988; Tatusov et al. 1994; Bücher et al. 1996). PSSMs are, quite simply, a numerical representation of a multiple sequence alignment, much like the multiple sequence alignments that will be discussed in Chapter 8. Embedded within a multiple sequence alignment is intrinsic sequence information that represents the common characteristics of that particular collection of sequences, frequently a protein family. By using a PSSM, one is able to use these embedded, common characteristics to find similarities between sequences with little or no absolute sequence identity, allowing for the identification and analysis of distantly related proteins. PSSMs are constructed by taking a multiple sequence alignment representing a protein family and then asking a series of questions, as follows.

- What residues are seen at each position of the alignment?
- How often does a particular residue appear at each position of the alignment?
- Are there positions that show absolute conservation?
- Can gaps be introduced anywhere in the alignment?

As soon as those questions are answered, the PSSM is constructed, and the numbers in the table now represent the multiple sequence alignment (Figure 3.13). The numbers within the PSSM reflect the probability of any given amino acid occurring at each position. The PSSM numbers also reflect the effect of a conservative or non-conservative substitution at each position in the alignment, much like the PAM or BLOSUM matrices do. This PSSM now can be used for comparison against single sequences, or in an iterative approach where newly found sequences can be incorporated into the original PSSM to find additional sequences that may be of interest.

The Method

Starting with a query sequence of interest, the PSI-BLAST process operates by taking a query protein sequence and performing a standard BLASTP search, as described above. This search produces a number of hits having E values better than a certain set threshold. These hits, along with the initial, single-query sequence, are used to construct a PSSM in an automated fashion. As soon as the PSSM is constructed, the PSSM then serves as the query for doing a new search against the target database, using the collective characteristics of the identified sequences to find new, related sequences. The process continues, round by round, either until the search converges (meaning that no new sequences were found in the last round) or until the limit on the number of iterations is reached.

Performing a PSI-BLAST Search

PSI-BLAST searches can be initiated by following the Protein BLAST link on the BLAST landing page (Figure 3.5). The search page shown in Figure 3.14 is identical to the one shown in the BLASTP example discussed earlier in this chapter. Here, the sequence of the human



Figure 3.13 Constructing a position-specific scoring matrix (PSSM). In the upper portion of the figure is a multiple sequence alignment of length 10. Using the criteria described in the text, the PSSM corresponding to this multiple sequence alignment is shown in the lower portion of the figure. Each row of the PSSM corresponds to a column in the multiple sequence alignment. Note that position 8 of the alignment always contains a threonine residue (T), whereas position 10 always contains a glycine (G). Looking at the corresponding scores in the matrix, in row 8, the threonine scores 150 points; in row 10, the glycine also scores 150 points. These are the highest values in the row, corresponding to the fact that the multiple sequence alignment shows absolute conservation at those positions. Now, consider position 9, where most of the sequences have a proline (P) at that position. In row 9 of the PSSM, the proline scores 89 points – still the highest value in the row, but not as high a score as would have been conferred if the proline residue was absolutely conserved across all sequences. The first column of the PSSM provides the deduced consensus sequence.

sex-determining protein SRY from UniProtKB/Swiss-Prot (Q05066) will be used as the query, using UniProtKB/Swiss-Prot as the target database and limiting returned results to human sequences. PSI-BLAST is selected in the Program Selection section and, as before, selected changes will be made to the default parameters (Figure 3.15). The maximum number of target sequences has been raised from 500 to 1000, as a safeguard in case a large number of sequences in UniProtKB/Swiss-Prot match the query. In addition, both the *E* value threshold and the PSI-BLAST threshold have been changed to 0.001, and filtering of low-complexity regions has been enabled. The query can now be issued as before by clicking on the blue “BLAST” button at the bottom of the page.

The results of the first round of the search are shown in Figure 3.16, with 31 sequences found in the first round (at the time of this writing). The structure of the hit list table is exactly as before, now containing two additional columns that are specific to PSI-BLAST. The first shows a column of check boxes that are all selected; this instructs the algorithm to use all the sequences to construct the first PSSM for this particular search. Keeping in mind that the first round of any PSI-BLAST search is simply a BLASTP search and that no PSSM has yet been constructed, the second column is blank. To run the next iteration of PSI-BLAST, simply click the “Go” button at the bottom of this section. At this point, the first PSSM is constructed based on a multiple sequence alignment of the sequences selected for inclusion, and the matrix is now used as the query against Swiss-Prot. The results of this second round are shown in Figure 3.17, with the final two columns indicating which sequences are to be used in constructing the new PSSM for the next round of searches, as well as which sequences were used to build the PSSM for the current round. Also note that a good number of the sequences are highlighted in yellow; here, 26 additional sequences that scored below the PSI-BLAST threshold in the first

The screenshot shows the NCBI BLAST web interface for a PSI-BLAST search. The browser address bar shows the URL: https://blast.ncbi.nlm.nih.gov/Blast.cgi?PROGRAM=blastp&PAGE_TYPE=BlastSearch&LINK_LOC=blasthome. The page header includes the NIH logo, "U.S. National Library of Medicine", and "NCBI National Center for Biotechnology Information". The main heading is "BLAST® >> blastp suite".

The "Align Sequences Protein BLAST" section contains the following fields and options:

- Enter Query Sequence:** A text box containing "Q05066". A "Clear" button is next to it. To the right, "Query subrange" has "From" and "To" input boxes.
- Or, upload file:** A "Browse..." button and the text "No file selected."
- Job Title:** A text box containing "Q05066:RecName: Full=Sex-determining region...". Below it is a prompt: "Enter a descriptive title for your BLAST search".
- Align two or more sequences**

The "Choose Search Set" section includes:

- Database:** A dropdown menu set to "UniProtKB/Swiss-Prot (swissprot)".
- Organism:** A text box containing "Homo sapiens (taxid:9606)". An "Exclude" checkbox is checked.
- Exclude:** A checked checkbox for "Models (XM/XP)" and an unchecked checkbox for "Uncultured/environmental sample sequences".
- Entrez Query:** An empty text box with a "Create custom database" link.

The "Program Selection" section shows the "Algorithm" dropdown set to "PSI-BLAST (Position-Specific Iterated BLAST)". Other options include "blastp (protein-protein BLAST)", "PHI-BLAST (Pattern Hit Initiated BLAST)", and "DELTA-BLAST (Domain Enhanced Lookup Time Accelerated BLAST)".

At the bottom, a large "BLAST" button is present. To its right, the text reads: "Search database UniProtKB/Swiss-Prot (swissprot) using PSI-BLAST (Position-Specific Iterated BLAST)". Below this is a checked checkbox for "Show results in a new window".

A note at the bottom right states: "Note: Parameter values that differ from the default are highlighted in yellow and marked with a diamond sign." The UniProtKB/Swiss-Prot database name and the PSI-BLAST algorithm name are highlighted in yellow and marked with a diamond sign.

Figure 3.14 Performing a PSI-BLAST search. See text for details.

round have now been pulled into the search results. This provides an excellent example of how PSSMs can be used to discover new relationships during each PSI-BLAST iteration, thereby making it possible to identify additional homologs that may not have been found using the standard BLASTP approach. Of course, the user should always check the *E* values and percent identities for all returned results before passing them through to the next round, unchecking inclusion boxes as needed. There may also be cases where prior knowledge would argue for removing some of the found sequences based on the descriptors. As with all computational methods, it is always important to keep biology in mind when reviewing the results.

BLAT

In response to the assembly needs of the Human Genome Project, a new nucleotide sequence alignment program called BLAT (for BLAST-Like Alignment Tool) was introduced (Kent 2002). BLAT is most similar to the MegaBLAST version of BLAST in that it is designed

Protein BLAST: Align two o... x

https://blast.ncbi.nlm.nih.gov/Blast.cgi?PROGRAM=blastp&PAGE_TYPE=BlastSearch&LINK_LOC=blasthome

100% Search

Algorithm parameters Note: Parameter values that differ from the default are highlighted in yellow and marked with a sign

General Parameters

Max target sequences 1000 Select the maximum number of aligned sequences to display

Short queries Automatically adjust parameters for short input sequences

Expect threshold 0.001

Word size 3

Max matches in a query range 0

Scoring Parameters

Matrix BLOSUM62

Gap Costs Existence: 11 Extension: 1

Compositional adjustments Conditional compositional score matrix adjustment

Filters and Masking

Filter Low complexity regions

Mask Mask for lookup table only Mask lower case letters

PSI/PHI/DELTA BLAST

Upload PSSM Browse... No file selected.

Optional

PSI-BLAST Threshold 0.001

Pseudocount 0

BLAST Search database UniProtKB/Swiss-Prot(swissprot) using PSI-BLAST (Position-Specific Iterated BLAST) Show results in a new window

Figure 3.15 Selecting algorithm parameters for a PSI-BLAST search. See text for details.

to rapidly align longer nucleotide sequences having more than 95% similarity. However, the BLAT algorithm uses a slightly different strategy than BLAST to achieve faster speeds. Before any searches are performed, the target databases are pre-indexed, keeping track of all non-overlapping 11-mers; this index is then used to find regions similar to the query sequence. BLAT is often used to find the position of a sequence of interest within a genome or to perform cross-species analyses.

As an example, consider a case where an investigator wishes to map a cDNA clone coming from the Cancer Genome Anatomy Project (CGAP) to the rat genome. The BLAT query page is shown in Figure 3.18, and the sequence of the clone of interest has been pasted into the sequence box. Above the sequence box are several pull-down menus that can be used to specify which genome should be searched (organism), which assembly should be used (usually, the most recent), and the query type (DNA, protein, translated DNA, or translated RNA). Once the appropriate choices have been made, the search is commenced by pressing the “Submit” button. The results of the query are shown in the upper panel of Figure 3.19; here, the hit with the highest score is shown at the top of the list, a match having 98.1% identity with the query sequence. More details on this hit can be found by clicking the “details” hyperlink, to the left of the entry. A long web page is then returned, providing information on the original query, the genomic sequence, and an alignment of the query against the found genomic sequence

NCBI Blast:Q05066:RecNa... x

https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Get&RID=G9VAXHC6015

120% Search

Descriptions

Run PSI-Blast iteration 2 with max 1000

Sequences producing significant alignments with E-value BETTER than threshold

Select: [All](#) [None](#) Selected:0

Alignments

Description	Max score	Total score	Query cover	E value	Ident	Accession	Select for PSI blast	Used to build PSSM
<input type="checkbox"/> RecName: Full=Sex-determining region Y protein; AltName: Full=Testis-determining	428	428	100%	9e-156	100%	Q05066.1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-2	128	128	68%	3e-36	46%	P48431.1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-3	129	129	37%	9e-36	74%	P41225.2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-21; AltName: Full=SOX-A	124	124	39%	4e-35	70%	Q9Y651.1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-14; AltName: Full=Protein SOX-28	120	120	39%	6e-34	67%	Q95416.1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-1	123	123	37%	1e-33	70%	Q00570.2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Protein SOX-15; AltName: Full=Protein SOX-12; AltName: Full=P	115	115	38%	7e-32	67%	Q60248.1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-11	105	105	39%	8e-27	58%	P35716.2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-7	104	104	55%	2e-26	46%	Q9BT81.1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-17	103	103	39%	3e-26	52%	Q9H6I2.1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-4	99.8	99.8	37%	1e-24	55%	Q06945.1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-18	99.0	99.0	38%	1e-24	53%	P35713.2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-12; AltName: Full=Protein SOX-22	96.7	96.7	39%	4e-24	54%	O15370.2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-8	96.7	96.7	36%	1e-23	52%	P57073.1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-5	96.7	96.7	40%	2e-23	51%	P35711.3	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-6	95.9	95.9	36%	4e-23	55%	P35712.3	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-9	95.5	95.5	36%	6e-23	52%	P48436.1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-10	94.7	94.7	41%	8e-23	46%	P56693.1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-13; AltName: Full=Islet cell antigen 12; f	93.6	93.6	36%	3e-22	55%	Q9UN79.3	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-30	78.2	78.2	41%	7e-17	49%	Q94993.1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Protein capicua homolog	53.5	53.5	62%	2e-08	28%	Q96RK0.2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=High mobility group protein B3; AltName: Full=High mobility group	45.1	45.1	33%	5e-06	31%	O15347.4	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor 7-like 1; AltName: Full=HMG box transcription	45.8	45.8	35%	6e-06	29%	Q9HCS4.1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor 7; Short=TCF-7; AltName: Full=T-cell-specific	45.1	45.1	44%	9e-06	28%	P36402.3	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Lymphoid enhancer-binding factor 1; Short=LEF-1; AltName: Full:	43.5	43.5	35%	3e-05	26%	Q9UJU2.1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=HMG box transcription factor BBX; AltName: Full=Bobby sox hom	43.5	43.5	36%	4e-05	31%	Q8WY36.1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor 7-like 2; AltName: Full=HMG box transcription	42.7	42.7	33%	6e-05	28%	Q9NQB0.2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=High mobility group protein B2; AltName: Full=High mobility group	40.0	40.0	36%	3e-04	28%	P26583.2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=HMG box-containing protein 1; AltName: Full=HMG box transcript	40.4	40.4	46%	3e-04	31%	Q60381.2	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=FACT complex subunit SSRP1; AltName: Full=Chromatin-specific	40.4	40.4	38%	4e-04	25%	Q08945.1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> RecName: Full=Nucleolar transcription factor 1; AltName: Full=Autoantigen NOR-	39.3	39.3	27%	0.001	33%	P17480.1	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Run PSI-Blast iteration 2 with max 1000

Figure 3.16 Results of the first round of a PSI-BLAST search. For each sequence found, the user is presented with the definition line from the corresponding UniProtKB/Swiss-Prot entry, the score value for the best high-scoring segment pair (HSP) alignment, the total of all scores across all HSP alignments, the percentage of the query covered by the HSPs, and the E value and percent identity for the best HSP alignment. The hyperlinked accession number allows for direct access to the source database record for that hit. Sequences whose "Select for PSI blast" box are checked will be used to calculate a position-specific scoring matrix (PSSM), and that PSSM then serves as the new "query" for the next round, the results of which are shown in Figure 3.17.

NCBI Blast: Q05066:RecNa... x

https://blast.ncbi.nlm.nih.gov/Blast.cgi

Select: All None Selected:0 Yellow: sequences scoring below threshold on previous iteration

Alignments Download GenPept Graphics Distance tree of results Multiple alignment

Description	Max score	Total score	Query cover	E value	Ident	Accession	Select for PSI blast	Used to build PSSM
<input type="checkbox"/> RecName: Full=Transcription factor SOX-2	183	183	70%	3e-57	45%	P48431.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Protein capicua homolog	163	163	64%	1e-46	27%	Q96RK0.2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-3	152	152	62%	3e-44	47%	P41225.2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-7	148	148	65%	4e-43	39%	Q9BT81.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-14; AltName: Full=Protein SOX-28	142	142	66%	2e-42	47%	Q95416.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-21; AltName: Full=SOX-A	136	136	45%	1e-39	63%	Q9Y651.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-11	136	136	47%	4e-38	49%	P35716.2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-17	132	132	39%	7e-37	52%	Q9H612.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-12; AltName: Full=Protein SOX-22	130	130	42%	8e-37	50%	Q15370.2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-4	131	131	41%	3e-36	50%	Q06945.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-10	130	130	68%	6e-36	31%	P56693.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-5	132	132	41%	6e-36	50%	P35711.3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Protein SOX-15; AltName: Full=Protein SOX-12; AltName: Full=Pro	125	125	48%	1e-35	55%	Q60248.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-1	127	127	39%	2e-35	68%	Q00570.2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-13; AltName: Full=Islet cell antigen 12; Alt	130	130	68%	3e-35	34%	Q9UN79.3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-9	129	129	42%	5e-35	46%	P48436.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-8	127	127	40%	6e-35	48%	P57073.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-6	129	129	41%	6e-35	49%	P35712.3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-18	125	125	41%	2e-34	49%	P35713.2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=HMG box-containing protein 1; AltName: Full=HMG box transcrip	112	112	47%	3e-29	29%	Q60381.2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor 7; Short=TCF-7; AltName: Full=T-cell-specific tr	105	105	49%	4e-27	26%	P36402.3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=FACT complex subunit SSRP1; AltName: Full=Chromatin-specific tr	105	105	42%	2e-26	24%	Q08945.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor SOX-30	104	104	41%	5e-26	49%	Q94993.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Lymphoid enhancer-binding factor 1; Short=LEF-1; AltName: Full=T	102	102	40%	7e-26	23%	Q9UJU2.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=HMG box transcription factor BBX; AltName: Full=Bobby sox homol	102	102	63%	2e-25	24%	Q8WY36.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=High mobility group protein B3; AltName: Full=High mobility group p	95.6	146	50%	1e-24	27%	O15347.4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor 7-like 1; AltName: Full=HMG box transcription fi	98.7	98.7	55%	5e-24	22%	Q9HCS4.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Transcription factor 7-like 2; AltName: Full=HMG box transcription fi	94.8	94.8	33%	1e-22	28%	Q9NQB0.2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=High mobility group protein B2; AltName: Full=High mobility group p	86.3	133	51%	5e-21	27%	P26583.2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=High mobility group protein B1; AltName: Full=High mobility group p	81.0	134	59%	6e-19	25%	P09429.3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> PUTATIVE PSEUDOGENE: RecName: Full=Putative high mobility group protein B1	75.2	120	51%	7e-17	24%	B2RPK0.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Nucleolar transcription factor 1; AltName: Full=Autoantigen NOR-9	72.5	167	59%	6e-15	23%	P17480.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=SWI/SNF-related matrix-associated actin-dependent regulator of ch	70.9	70.9	39%	1e-14	23%	Q9P0W2.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Protein polybromo-1; Short=hPB1; AltName: Full=BRG1-associat	70.2	70.2	57%	3e-14	17%	Q86U86.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Nuclear autoantigen Sp-100; AltName: Full=Nuclear dot-associated	69.0	69.0	39%	9e-14	23%	P23497.3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=TOX high mobility group box family member 4; AltName: Full=Epide	64.8	64.8	35%	2e-12	21%	Q94842.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=High mobility group protein 20A; AltName: Full=HMG box-containin	62.8	62.8	44%	9e-12	18%	Q9NP66.1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=HMG domain-containing protein 4; AltName: Full=HMG box-contain	63.2	63.2	62%	9e-12	25%	Q9UGU5.2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> RecName: Full=Thymocyte selection-associated high mobility group box protein IQ	61.7	61.7	78%	3e-11	16%	Q94900.3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 3.17 Results of the second round of a PSI-BLAST search. New sequences identified through the use of the position-specific scoring matrix (PSSM) calculated based on the results shown in Figure 3.16 are highlighted in yellow. Check marks in the right-most column indicate which sequences were used to build the PSSM producing these results.

Rat BLAT Search

BLAT Search Genome

Genome: Assembly: Query type: Sort output: Output type:

```
>CB312815.1 NICHDRr Pit1 Rattus norvegicus cDNA clone sequence
GGGGCTCTCGCTGGCCTGTGTCTCAGAAGCTGCTTCTCCACCCTCTTCCTTGGAATTCCTAAACTCTC
TACCTCTGGTTCATGTTTCGCTCTCTGGATAGTCTGTGCAATGAGCCCTTAAGGAATATTGCAATGA
GCTATAGAGTTGTGAGCTCGGTAGGCAAGCCCTGCACTGGGACAGCAAGGAAATTCATTCGATCTC
GCTCCTAAGTCACAGGTTATCCAGAGCCCACTTACCACAAGAGACAGCCTCTCCCCATCCCTAGGAAA
CAGTAGAGCTTAGGAAAATGAATGACTCCACCACATTCAAGAGGCTTCAAATGTATACTTGGCATTCTC
GATTCAGTCTGAAAATTCGTCCCTTAGTCTGGGGAAAATAAGAAATGGAGTTACACCTTGTCAATTA
AAAAACATTGAATTAAGAGAAAATGGAAAATCATGCCACATAAAACATGTATGGAAGTGTTCATGTTTT
GATCATGGGGGGGATATAGCTCAGTCATGGAGTGTTCATAGCAATGTGCATAATCCGAGGTTCAAGC
CCCAGCACGAAAAAGAGAAAACGGGAGGAGTGGAGGCATTCACAGCAGCGTTTTTCAGTATAGGCGCAAAG
GGGAAGGAGTTTAAACACCTACTGAGGGAATGGATAAGCGGAGTGCCTTGTCTATACTCGGGATGGCT
AGTCATCACGTAAGAAAAGTTTGGAAAATGATAAAATACCAATGGGATGGATCCCTTTAAACCATCC
```

Paste in a query sequence to find its location in the the genome. Multiple sequences may be searched if separated by lines starting with '>' followed by the sequence name.

File Upload: Rather than pasting a sequence, you can choose to upload a text file containing the sequence.

Upload sequence: No file selected.

Only DNA sequences of 25,000 or fewer bases and protein or translated sequence of 10000 or fewer letters will be processed. Up to 25 sequences can be submitted at the same time. The total limit for multiple sequence submissions is 50,000 bases or 25,000 letters.

For locating PCR primers, use [In-Silico PCR](#) for best results instead of BLAT.

Figure 3.18 Submitting a BLAT query. A rat clone from the Cancer Genome Anatomy Project Tumor Gene Index (CB312815) is the query. The pull-down menus at the top of the page can be used to specify which genome should be searched (organism), which assembly should be used (usually, the most recent), and the query type (DNA, protein, translated DNA, or translated RNA). The “I’m feeling lucky” button returns only the highest scoring alignment and provides a direct path to the UCSC Genome Browser.

(Figure 3.19, bottom panel). The genomic sequence here is labeled chr5, meaning that the query corresponds to a region of rat chromosome 5. Matching bases in the cDNA and genomic sequences are colored in dark blue and are capitalized. Lighter blue uppercase bases mark the boundaries of aligned regions and often signify splice sites. Gaps and unaligned regions are indicated by lower case black type. In the Side by Side Alignment, exact matches are indicated by the vertical line between the two sequences. Clicking on the “browser” hyperlink in the upper panel of Figure 3.19 would take the user to the UCSC Genome Browser, where detailed information about the genomic assembly in this region of rat chromosome 5 (specifically, at 5q31) can be obtained (cf. Chapter 4).

FASTA

While the most commonly used technique for detecting similarity between sequences is BLAST, it is not the only heuristic method that can be used to rapidly and accurately compare sequences with one another. In fact, the first widely used program designed for database similarity searching was FASTA (Lipman and Pearson 1985; Pearson and Lipman 1988; Pearson 2000). Like BLAST, FASTA enables the user to rapidly compare a query sequence against large databases, and various versions of the program are available (Table 3.3). In addition to the main implementations, a variety of specialized FASTA versions are available, described in detail

Rat BLAT Results

BLAT Search Results

Go back to [chr1:80608553-80639261](#) on the Genome Browser.

ACTIONS	QUERY	SCORE	START	END	QSIZE	IDENTITY	CHRO	STRAND	START	END	SPAN
browser details	CB312815.1	710	1	733	768	98.1%	5	+	100687130	100687854	725
browser details	CB312815.1	30	502	544	768	77.2%	10	+	13863960	13863997	38
browser details	CB312815.1	29	501	537	768	89.2%	10	-	111879996	111880032	37
browser details	CB312815.1	29	501	537	768	89.2%	13	+	16666056	16666092	37
browser details	CB312815.1	26	496	529	768	93.4%	10	+	76193711	76193746	36
browser details	CB312815.1	24	501	530	768	96.2%	11	+	39710506	39710536	31
browser details	CB312815.1	21	202	222	768	100.0%	17	-	27661921	27661941	21
browser details	CB312815.1	21	16	36	768	100.0%	16	-	49891836	49891856	21
browser details	CB312815.1	21	502	522	768	100.0%	10	-	84776773	84776793	21
browser details	CB312815.1	21	502	532	768	83.9%	5	+	98956576	98956606	31
browser details	CB312815.1	21	552	574	768	95.7%	1	+	165557081	165557103	23
browser details	CB312815.1	20	424	443	768	100.0%	11	-	47244665	47244684	20
browser details	CB312815.1	20	442	461	768	100.0%	1	-	230070250	230070269	20
browser details	CB312815.1	20	508	527	768	100.0%	1	-	59335112	59335131	20
browser details	CB312815.1	20	508	527	768	100.0%	1	+	68449584	68449603	20

[Missing a match?](#)

Alignment of CB312815.1

[CB312815.1](#)
[Rat chr5](#)
[block1](#)
[together](#)

Genomic chr5 :

```

cttgaagaa ggtaactata cattaatata gagccctctt tttctttgca 100687079
ggcccaggac acacaggacg gatgtttcca agtcaactcca gggacagcat 100687129
GaGGCTCTCG CTGGCCTGTG TCTCAGAAGC TGCTTTCCTCC ACCTCTTCCT 100687179
TGGAATTTC CTAACCTCTC TACCTCTGGT TCATGTTCCG TCTCTGGAT 100687229
AGTCTGTGTG CAATGAGCCC TTAAGGAAT ATTGCAATGA GCTATAAGAG 100687279
TTGTGAGCCT CCGGTAGGCC AGGCCGTCAC TGGACAGCA AAGGAAATTT 100687329
CATTGCATCT GCTCCTAAGT CACAGGTTAT CCAGAGCCCA CTTTACCCCA 100687379
AGAGACAGCC TCTCCCCCAT CCCTAGGAAA CAGTAGAGCT TAGGAAAATG 100687429
AATGACTCCA CCACATTCAA GAGGCTTCAA ATTTATACT TGGCATTTC 100687479
GATTTCAAGT CTGAATTTCT GTCCCTTAGT CGTGGGGAAA ATAAGAAAATG 100687529
GAGTTACACC TTGTCATTTA AAAAACCAAT GAATTAAGAG AAATGGAAAA 100687579
TCATGCCACC ATAAACATG TATGGAAGTG TTCATGTTTT GATCATGGCG 100687629
GGGATATAG CTCAGTCATG GAGTGTTCG ATAGCAATG GCATAATCCG 100687679
AGGTTCAAGC CCCAGCACCG AAAAAGAGAA gCGGGAGGAG TGGAGGCATT 100687729
CACAGCAGCC TTTTCAGTAT AGGCAGAAAG GGGGAAGGAT TTAAACACCT 100687779
ACTGAGGAAT GGATAAGCGG AGTGCTTGT CTATACTCGG gat@CTAGTC 100687829
ATCAgtAGAA AAGTTTGAAT TGATAgatac gatggatgat cccctaaaca 100687879
tctcctaag taacagaggtc agcctagga ggcgtatgtt ccatatttct 100687929
gggatatatg ggcctaccga tgca
  
```

Side by Side Alignment

```

00000001 gggctctcgtgcccctgtgtctcagaagctgtttctcacccttctct 00000050
>>>>>>>> |||
100687130 gaggctctcgtgcccctgtgtctcagaagctgtttctcacccttctct 100687179
>>>>>>>> |||

00000051 tgtgaatttctaaactctctaccctcggttcattgctctctctggat 00000100
>>>>>>>> |||
100687180 tgtgaatttctaaactctctaccctcggttcattgctctctctggat 100687229
>>>>>>>> |||

00000101 agctgtgtgcaatgagcccttaaaaggaattgcaatgagctataagag 00000150
>>>>>>>> |||
100687230 agctgtgtgcaatgagcccttaaaaggaattgcaatgagctataagag 100687279
>>>>>>>> |||

00000151 ttgtgagcctgcgtaggcaaggcctgactgggacagaaaggaattt 00000200
>>>>>>>> |||
100687280 ttgtgagcctgcgtaggcaaggcctgactgggacagaaaggaattt 100687329
>>>>>>>> |||

00000201 cattgcattctcctcctaagtcacaggttatccagagccactttacccca 00000250
>>>>>>>> |||
100687330 cattgcattctcctcctaagtcacaggttatccagagccactttacccca 100687379
>>>>>>>> |||

00000251 agagacagcctctcccccaccctaggaacagtagagcttaggaaaatg 00000300
>>>>>>>> |||
100687380 agagacagcctctcccccaccctaggaacagtagagcttaggaaaatg 100687429
>>>>>>>> |||

00000301 aatgactccaccacattcaagaggttcaaaattgtataacttggcatttct 00000350
>>>>>>>> |||
100687430 aatgactccaccacattcaagaggttcaaaattgtataacttggcatttct 100687479
>>>>>>>> |||
  
```

Figure 3.19 Results of a BLAT query. Based on the query submitted in Figure 3.18, the highest scoring hit is to a sequence on chromosome 5 rat genome having 98.1% sequence identity. Clicking on the “details” hyperlink brings the user to additional information on the found sequence, shown in the lower panel. Matching bases in the cDNA and genomic sequences are colored in dark blue and are capitalized. Lighter blue uppercase bases mark the boundaries of aligned regions and often signify splice sites. Gaps are indicated by lowercase black type. In the side-by-side alignment, exact matches are indicated by the vertical line between the sequences.

Table 3.3 Main FASTA algorithms.

Program	Query	Database	Corresponding BLAST Program
FASTA	Nucleotide	Nucleotide	BLASTN
	Protein	Protein	BLASTP
FASTX/FASTY	DNA	Protein	BLASTX
TFASTYX/TFASTY	Protein	Translated DNA	TBLASTN

in Pearson (2016). An interesting historical note is that the FASTA format for representing nucleotide and protein sequences originated with the development of the FASTA algorithm.

The Method

The FASTA algorithm can be divided into four major steps. In the first step, FASTA determines all overlapping words of a certain length both in the query sequence and in each of the sequences in the target database, creating two lists in the process. Here, the word length parameter is called *ktup*, which is the equivalent of *W* in BLAST. These lists of overlapping words are compared with one another in order to identify any words that are common to the two lists. The method then looks for word matches that are in close proximity to one another and connects them to each other (intervening sequence included), without introducing any gaps. This can be represented using a dotplot format (Figure 3.20a). Once this initial round of connections are made, an initial score ($init_1$) is calculated for each of the regions of similarity.

In step 2, only the 10 best regions for a given pairwise alignment are considered for further analysis (Figure 3.20b). FASTA now tries to join together regions of similarity that are close to each other in the dotplot but that do not lie on the same diagonal, with the goal of extending the overall length of the alignment (Figure 3.20c). This means that insertions and deletions are now allowed, but there is a joining penalty for each of the diagonals that are connected. The net score for any two diagonals that have been connected is the sum of the score of the original diagonals, less the joining penalty. This new score is referred to as $init_n$.

In step 3, FASTA ranks all of the resulting diagonals, and then further considers only the “best” diagonals in the list. For each of the best diagonals, FASTA uses a modification of the Smith–Waterman algorithm (1981) to come up with the optimal pairwise alignment between the two sequences being considered. A final, optimal score (*opt*) is calculated on this pairwise alignment.

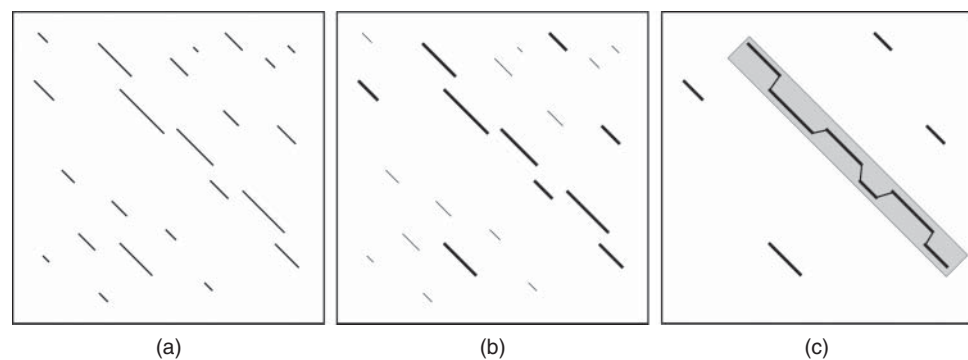


Figure 3.20 The FASTA search strategy. (a) Once FASTA determines words of length *ktup* common to the query sequence and the target sequence, it connects words that are close to each other, and these are represented by the diagonals. (b) After an initial round of scoring, the top 10 diagonals are selected for further analysis. (c) The Smith–Waterman algorithm is applied to yield the optimal pairwise alignment between the two sequences being considered. See text for details.

In the fourth and final step, FASTA assesses the significance of the alignments by estimating what the anticipated distribution of scores would be for randomly generated sequences having the same overall composition (i.e. sequence length and distribution of amino acids or nucleotides). Based on this randomization procedure and on the results from the original query, FASTA calculates an expectation value E (similar to the BLAST E value), which, as before, represents the probability that a reported hit has occurred purely by chance.

Running a FASTA Search

The University of Virginia provides a web front-end for issuing FASTA queries. Various protein and nucleotide databases are available, and up to two databases can be selected for use in a single run. From this page, the user can also specify the scoring matrix to be used, gap and extension penalties, and the value for *ktup*. The default values for *ktup* are 2 for protein-based searches and 6 for nucleotide-based searches; lowering the value of *ktup* increases the sensitivity of the run, at the expense of speed. The user can also limit the results returned to particular E values.

The results returned by a FASTA query are in a significantly different format than those returned by BLAST. Consider a FASTA search using the sequence of histone H2B.3 from the highly regenerative cnidarian *Hydractinia*, one of four novel H2B variants used in place of protamines to compact sperm DNA (KX622131.1; Török et al. 2016), as the query. The first part of the FASTA output resulting from a search using BLOSUM62 as the scoring matrix and Swiss-Prot as the target database is shown in Figure 3.21, summarizing the results as a histogram. The histogram is intended to convey the distribution of all similarity scores computed in the course of this particular search. The first column represents bins of similarity scores, with the scores increasing as one moves down the page. The second column gives the actual number of sequences observed to fall into each one of these bins. This count is also represented by the length of each of the lines in the histogram, with each of the equals signs representing a certain number of sequences; in the figure, each equals sign corresponds to 130 sequences from UniProtKB/Swiss-Prot. The third column of numbers represents how many sequences would be expected to fall into each one of the bins; this is indicated by the asterisks in the histogram. The hit list would immediately follow, and a portion of the hit list for this search is shown in Figure 3.22. Here, the accession number and partial definition line for each hit is given, along with its optimal similarity score (*opt*), a normalized score (*bit*), the expectation value (E), percent identity and similarity figures, and the aligned length. Not shown here are the individual alignments of each hit to the original query sequence, which would be found by further scrolling down in the output. In the pairwise alignments, exact matches are indicated by a colon, while conservative substitutions are indicated by a dot.

Statistical Significance of Results

As before, the E values from a FASTA search represent the probability that a hit has occurred purely by chance. Pearson (2016) puts forth the following guidelines for inferring homology from protein-based searches, which are slightly different than those previously described for BLAST: an E value $< 10^{-6}$ almost certainly implies homology. When $E < 10^{-3}$, the query and found sequences are almost always homologous, but the user should guarantee that the highest scoring unrelated sequence has an E value near 1.

Comparing FASTA and BLAST

Since both FASTA and BLAST employ rigorous algorithms to find sequences that are statistically (and hopefully biologically) relevant, it is logical to ask which one of the methods is the better choice. There actually is no good answer to the question, since both of the methods

```

Query: TMP.q
1>>>KX622131.1 Hydractinia echinata histone H2B.3 mRNA, complete cds - 474 aa
Library: Swissprot (550K)
198069095 residues in 553474 sequences

    opt      E( )
< 40    31    0:=
42      0     0:
44      0     1:*
46      2    14:*
48     62    83:*
50    102   315:= *
52    329   839:=== *
54   1572  1699:===== *
56   2748  2778:===== *
58   2860  3841:===== *
60   3822  4664:===== *
62   7781  5118:===== *
64   6960  5192:===== *
66   5775  4955:===== *
68   4804  4510:===== *
70   3457  3958:===== *
72   2876  3376:===== *
74   2291  2817:===== *
76   2006  2310:===== *
78   1585  1870:===== *
80   1198  1499:===== *
82    952  1191:===== *
84    731   941:===== *
86    581   740:===== *
88    512   580:===== *
90    464   453:===== *
92    340   353:===== *
94    260   275:===== *
96    227   214:===== *
98    197   166:===== *
100   144   127:===== *
102   138   101:===== *
104    86    78:*
106   105    60:*
108    62    47:*
110    68    36:*
112    58    28:*
114    45    22:*
116    45    17:*
118    38    13:*
120    30    10:*
122    41     8:*
124    34     6:*
126    22     5:*
128    31     4:*
130    38     3:*
132    11     2:*
134    16     2:*
136    15     1:*
138    13     1:*
>140  278     1:=====

                                one = represents 130 library sequences

                                inset = represents 6 library sequences

```

Figure 3.21 Search summary from a protein–protein FASTA search, using the sequence of histone H2B.3 from *Hydractinia echinata* (KX622131.1; Török et al. 2016) as the query and BLOSUM62 as the scoring matrix. The header indicates that the query is against the Swiss-Prot database. The histogram indicates the distribution of all similarity scores computed for this search. The left-most column provides a normalized similarity score, and the column marked `opt` gives the number of sequences with that score. The column marked `E()` gives the number of sequences expected to achieve the score in the first column. In this case, each equals sign in the histogram represents 130 sequences in Swiss-Prot. The asterisks in each row indicate the expected, random distribution of hits. The inset is a magnified version of the histogram in that region.

The best scores are:						opt	bits	E(553474)	% id	% sim	alen	
sp	Q7Z5P9	MUC19_HUMAN	Mucin-19 OS=Homo sapiens GN=MUC19	(8384)	232	84.5	8.1e-14	0.291	0.573	508	align	
+-					178	67.3	1.2e-08	0.292	0.590	517	align	
+-					178	67.3	1.2e-08	0.284	0.557	497	align	
+-					171	65.0	5.8e-08	0.299	0.591	499	align	
sp	A6BM72	MEG11_HUMAN	Multiple epidermal growth factor-	(1044)	222	80.5	1.6e-13	0.244	0.439	586	align	
sp	Q80T91	MEG11_MOUSE	Multiple epidermal growth factor-	(1091)	208	76.1	3.6e-12	0.244	0.451	525	align	
sp	Q80V70	MEGF6_MOUSE	Multiple epidermal growth factor-	(1572)	191	70.8	2e-10	0.265	0.420	581	align	
+-					130	51.4	0.00014	0.249	0.409	462	align	
sp	O75095	MEGF6_HUMAN	Multiple epidermal growth factor-	(1541)	182	67.9	1.4e-09	0.267	0.448	574	align	
+-					122	48.8	0.00083	0.252	0.421	432	align	
sp	P34504	YMV2_CAEEL	Uncharacterized protein K04H4.2 OS	(1463)	171	64.4	1.6e-08	0.273	0.501	381	align	
sp	O88281	MEGF6_RAT	Multiple epidermal growth factor-li	(1574)	171	64.4	1.7e-08	0.266	0.431	612	align	
sp	Q9W7R4	TEN3_DANRE	Teneurin-3 OS=Danio rerio GN=tenm3	(2590)	155	59.5	8.3e-07	0.239	0.435	322	align	
sp	Q5VY43	PEAR1_HUMAN	Platelet endothelial aggregation	(1037)	151	57.9	1e-06	0.239	0.404	564	align	
sp	Q9WTS6	TEN3_MOUSE	Teneurin-3 OS=Mus musculus GN=Tenm	(2715)	149	57.6	3.2e-06	0.241	0.465	282	align	
sp	Q9W7R3	TEN4_DANRE	Teneurin-4 OS=Danio rerio GN=tenm4	(2824)	149	57.6	3.3e-06	0.245	0.480	323	align	
sp	P10079	FBP1_STRPU	Fibropellin-1 OS=Strongylocentrotu	(1064)	145	56.0	3.9e-06	0.236	0.446	554	align	
sp	Q9P273	TEN3_HUMAN	Teneurin-3 OS=Homo sapiens GN=TENM	(2699)	143	55.7	1.2e-05	0.224	0.455	312	align	
sp	Q9UM47	NOTC3_HUMAN	Neurogenic locus notch homolog pr	(2321)	138	54.0	3.3e-05	0.219	0.459	521	align	
+-					117	47.4	0.0034	0.241	0.394	315	align	
sp	A8XMW6	CED1_CAEER	Cell death abnormality protein 1 O	(1134)	135	52.8	3.7e-05	0.255	0.441	478	align	
sp	P21849	TSA4_GIAIN	Major surface-labeled trophozoite	(713)	133	52.0	4.1e-05	0.258	0.485	462	align	
sp	Q9W0A0	DRPR_DROME	Protein draper OS=Drosophila melan	(1031)	132	51.8	6.8e-05	0.243	0.414	548	align	
sp	P17053	G168_PARPR	G surface protein, allelic form 16	(2704)	135	53.1	7.1e-05	0.273	0.473	410	align	

Figure 3.22 Hit list for the protein–protein FASTA search described in Figure 3.21. Only the first 18 hits are shown. For each hit, the accession number and partial definition line for the hit is provided. The column marked *opt* gives the raw similarity score, the column marked *bits* gives a normalized bit score (a measure of similarity between the two sequences), and the column marked *E* gives the expectation value. The percentage columns indicate percent identity and percent similarity, respectively. The *alen* column gives the total aligned length for each hit. The +- characters shown at the beginning of some lines indicate that more than one alignment was found between the query and subject; in the case of the first hit (Q7Z5P9), four alignments were returned. The *align* link at the end of each row takes the user to the alignment for that hit (not shown).

bring significant strengths to the table. Summarized below are some of the fine points that distinguish the two methods from one another.

- FASTA begins the search by looking for exact matches of words, while BLAST allows for conservative substitutions in the first step.
- BLAST allows for automatic masking of sequences, while FASTA does not.
- FASTA will return one and only one alignment for a sequence in the hit list, while BLAST can return multiple results for the same sequence, each result representing a distinct HSP.
- Since FASTA uses a version of the more rigorous Smith–Waterman alignment method, it generally produces better final alignments and is more apt to find distantly related sequences than BLAST. For highly similar sequences, their performance is fairly similar.
- When comparing translated DNA sequences with protein sequences or vice versa, FASTA (specifically, FASTX/FASTY for translated DNA → protein and TFASTX/TFASTY for protein → translated DNA) allows for frameshifts.
- BLAST runs faster than FASTA, since FASTA is more computationally intensive.

Several studies have attempted to answer the “which method is better” question by performing systematic analyses with test datasets (Pearson 1995; Agarawal and States 1998; Chen 2003). In one such study, Brenner et al. (1998) performed tests using a dataset derived from already known homologies documented in the Structural Classification of Proteins database (SCOP; Chapter 12). They found that FASTA performed better than BLAST in finding relationships between proteins having >30% sequence identity, and that the performance of all methods declines below 30%. Importantly, while the statistical values reported by BLAST slightly underestimated the true extent of errors when looking for known relationships, they found that BLAST and FASTA (with *ktup* = 2) were both able to detect most known relationships, calling them both “appropriate for rapid initial searches.”

Summary

The ability to perform pairwise sequence alignments and interpret the results from such analyses has become commonplace for nearly all biologists, no longer being a technique employed solely by bioinformaticians. With time, these methods have undergone a continual evolution, keeping pace with the types and scale of data that are being generated both in individual laboratories and by systematic, organismal sequencing projects. As with all computational techniques, the reader should have a firm grasp of the underlying algorithm, always keeping in mind the algorithm's capabilities and limitations. Intelligent use of the tools presented in this chapter can lead to powerful and interesting biological discoveries, but there have also been many cases documented where improper use of the tools has led to incorrect biological conclusions. By understanding the methods, users can optimally use them and end up with a better set of results than if these methods were treated simply as a "black box." As biology is increasingly undertaken in a sequence-based fashion, using sequence data to underpin the design and interpretation of experiments, it becomes increasingly important that computational results, such as those generated using BLAST and FASTA, are cross-checked in the laboratory, against the literature, and with additional computational analyses to ensure that any conclusions drawn not only make biological sense but also are actually correct.

Internet Resources

BLAST

European Bioinformatics Institute (EBI)

www.ebi.ac.uk/blastall

National Center for Biotechnology Information (NCBI)

blast.ncbi.nlm.nih.gov

BLAST-Like Alignment Tool (BLAT)

genome.ucsc.edu/cgi-bin/hgBlat

NCBI Conserved Domain Database (CDD)

ncbi.nlm.nih.gov/cdd

Cancer Genome Anatomy Project (CGAP)

ocg.cancer.gov/programs/cgap

FASTA

EBI

www.ebi.ac.uk/Tools/sss/fasta

University of Virginia

fasta.bioch.virginia.edu

RefSeq

ncbi.nlm.nih.gov/refseq

Structural Classification of Proteins (SCOP)

scop.berkeley.edu

Swiss-Prot

www.uniprot.org

Further Reading

Altschul, S.F., Boguski, M.S., Gish, W., and Wootton, J.C. (1994). Issues in searching molecular sequence databases. *Nat. Genet.* 6: 119–129. A review of the issues that are of importance in using sequence similarity search programs, including potential pitfalls.

Fitch, W. (2000). Homology: a personal view on some of the problems. *Trends Genet.* 16: 227–231. A classic treatise on the importance of using precise terminology when describing the relationships between biological sequences.

Henikoff, S. and Henikoff, J.G. (2000). Amino acid substitution matrices. *Adv. Protein Chem.* 54: 73–97. A comprehensive review covering the factors critical to the construction of protein scoring matrices.

Koonin, E. (2005). Orthologs, paralogs, and evolutionary genomics). *Annu. Rev. Genet.* 39: 309–338. An in-depth explanation of orthologs, paralogs, and their subtypes, with a discussion of their evolutionary origin and strategies for their detection.

- Pearson, W.R. (2016). Finding protein and nucleotide similarities with FASTA. *Curr. Protoc. Bioinf.* 53: 3.9.1–3.9.23. An in-depth discussion of the FASTA algorithm, including worked examples and additional information regarding run options and use scenarios.
- Wheeler, D.G. (2003). Selecting the right protein scoring matrix. *Curr. Protoc. Bioinf.* 1: 3.5.1–3.5.6. A discussion of PAM, BLOSUM, and specialized scoring matrices, with guidance regarding the proper choice of matrices for particular types of protein-based analyses.

References

- Agarawal, P. and States, D.J. (1998). Comparative accuracy of methods for protein similarity search. *Bioinformatics.* 14: 40–47.
- Altschul, S.F. (1991). Amino acid substitution matrices from an information theoretic perspective. *J. Mol. Biol.* 219: 555–565.
- Altschul, S.F. and Koonin, E.V. (1998). Iterated profile searches with PSI-BLAST: a tool for discovery in protein databases. *Trends Biochem. Sci.* 23: 444–447.
- Altschul, S.F., Gish, W., Miller, W. et al. (1991). Basic local alignment search tool. *J. Mol. Biol.* 215: 403–410.
- Altschul, S.F., Madden, T.L., Schäffer, A.A. et al. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* 25: 3389–3402.
- Brenner, S.E., Chothia, C., and Hubbard, T.J.P. (1998). Assessing sequence comparison methods with reliable structurally identified evolutionary relationships. *Proc. Natl. Acad. Sci. USA.* 95: 6073–6078.
- Bücher, P., Karplus, K., Moeri, N., and Hofmann, K. (1996). A flexible motif search technique based on generalized profiles. *Comput. Chem.* 20: 3–23.
- Chen, Z. (2003). Assessing sequence comparison methods with the average precision criterion. *Bioinformatics.* 19: 2456–2460.
- Dayhoff, M.O., Schwartz, R.M., and Orcutt, B.C. (1978). A model of evolutionary change in proteins. In: *Atlas of Protein Sequence and Structure*, vol. 5 (ed. M.O. Dayhoff), 345–352. Washington, DC: National Biomedical Research Foundation.
- Doolittle, R.F. (1981). Similar amino acid sequences: chance or common ancestry. *Science* 214: 149–159.
- Doolittle, R.F. (1989). Similar amino acid sequences revisited. *Trends Biochem. Sci.* 14: 244–245.
- Gonnet, G.H., Cohen, M.A., and Benner, S.A. (1992). Exhaustive matching of the entire protein sequence database. *Proteins.* 256: 1443–1445.
- Gribskov, M., McLachlan, A.D., and Eisenberg, D. (1987). Profile analysis: detection of distantly-related proteins. *Proc. Natl. Acad. Sci. USA.* 84: 4355–4358.
- Henikoff, S. and Henikoff, J.G. (1991). Automated assembly of protein blocks for database searching. *Nucleic Acids Res.* 19: 6565–6572.
- Henikoff, S. and Henikoff, J.G. (1992). Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA.* 89: 10915–10919.
- Henikoff, S. and Henikoff, J.G. (1993). Performance evaluation of amino acid substitution matrices. *Proteins Struct. Funct. Genet.* 17: 49–61.
- Henikoff, S. and Henikoff, J.G. (2000). Amino acid substitution matrices. *Adv. Protein Chem.* 54: 73–97.
- Jones, D.T., Taylor, W.R., and Thornton, J.M. (1992). The rapid generation of mutation data matrices from protein sequences. *Comput. Appl. Biosci.* 8: 275–282.
- Karlin, S. and Altschul, S.F. (1990). Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc. Natl. Acad. Sci. USA.* 87: 2264–2268.
- Kent, W.J. (2002). BLAT: the BLAST-like alignment tool. *Genome Res.* 12: 656–664.
- Lipman, D.J. and Pearson, W.R. (1985). Rapid and sensitive protein similarity searches. *Science.* 227: 1435–1441.

- Ma, B., Tromp, J., and Li, M. (2002). PatternHunter: faster and more sensitive homology search. *Bioinformatics*. 18: 440–445.
- Pearson, W.R. (1995). Comparison of methods for searching protein sequence databases. *Protein Sci*. 4: 1145–1160.
- Pearson, W.R. (2000). Flexible sequence similarity searching with the FASTA3 program package. *Methods Mol. Biol.* 132: 185–219.
- Pearson, W.R. (2016). Finding protein and nucleotide similarities with FASTA. *Curr. Protoc. Bioinf.* 53: 3.9.1–3.9.23.
- Pearson, W.R. and Lipman, D.J. (1988). Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA*. 85: 2444–2448.
- Rost, B. (1999). Twilight zone of protein sequence alignments. *Protein Eng.* 12: 85–94.
- Ryan, J.F., Pang, K., Schnitzler, C.E. et al., and NISC Comparative Sequencing Program. (2013). The genome of the ctenophore *Mnemiopsis leidyi* and its implications for cell type evolution. *Science*. 342: 1242592.
- Schneider, T.D., Stormo, G.D., Gold, L., and Ehrenfeucht, A. (1986). Information content of binding sites on nucleotide sequences. *J. Mol. Biol.* 188: 415–431.
- Schnitzler, C.E., Simmons, D.K., Pang, K. et al. (2014). Expression of multiple Sox genes through embryonic development in the ctenophore *Mnemiopsis leidyi* is spatially restricted to zones of cell proliferation. *EvoDevo*. 5: 15.
- Smith, T.F. and Waterman, M.S. (1981). Identification of common molecular subsequences. *J. Mol. Biol.* 147: 195–197.
- Staden, R. (1988). Methods to define and locate patterns of motifs in sequences. *Comput. Appl. Biosci.* 4: 53–60.
- Tatusov, R.L., Altschul, S.F., and Koonin, E.V. (1994). Detection of conserved segments in proteins: iterative scanning of sequence databases with alignment blocks. *Proc. Natl. Acad. Sci. USA*. 91: 12091–12095.
- Tatusova, T.A. and Madden, T.L. (1999). BLAST 2 Sequences, a new tool for comparing protein and nucleotide sequences. *FEMS Microbiol. Lett.* 174: 247–250.
- Török, A., Schiffer, P.H., Schintzler, C.E. et al. (2016). The cnidarian *Hydractinia echinata* employs canonical and highly adapted histones to pack its DNA. *Epigenet. Chromatin*. 9: 36.
- Vogt, G., Etzold, T., and Argos, P. (1995). An assessment of amino acid exchange matrices in aligning protein sequences: the twilight zone revisited. *J. Mol. Biol.* 249: 816–831.
- Wheeler, D.G. (2003). Selecting the right protein scoring matrix. *Curr. Protoc. Bioinf.* 1: 3.5.1–3.5.6.
- Wootton, J.C. and Federhen, S. (1993). Statistics of local complexity in amino acid sequences and sequence databases. *Comput. Chem.* 17: 149–163.
- Zhang, Z., Schwartz, S., Wagner, L., and Miller, W. (2000). A greedy algorithm for aligning DNA sequences. *J. Comput. Biol.* 7: 203–214.

This chapter was written by Dr. Andreas D. Baxeavanis in his private capacity. No official support or endorsement by the National Institutes of Health or the United States Department of Health and Human Services is intended or should be inferred.

4

Genome Browsers

Tyra G. Wolfsberg

Introduction

The first complete sequence of a eukaryotic genome – that of *Saccharomyces cerevisiae* – was published in 1996 (Goffeau et al. 1996). The chromosomes of this organism, which range in size from 270 to 1500 kb, presented an immediate challenge in data management, as the upper limit for single database entries in GenBank at the time was 350 kb. To better manage the yeast genome sequence, as well as other chromosome and genome-length sequences being deposited into GenBank around that time, the National Center for Biotechnology Information (NCBI) at the National Institutes of Health (NIH) established the Genomes division of Entrez (Benson et al. 1997). Entries in this division were organized around a reference sequence onto which all other sequences from that organism were aligned. As these reference sequences have no size limit, “virtual” reference sequences of large genomes or chromosomes could be assembled from shorter GenBank sequences. For partially sequenced chromosomes, NCBI developed methods to integrate genetic, physical, and cytogenetic maps onto the framework of the whole chromosome. Thus, Entrez Genomes was able to provide the first graphical views of large-scale genomic sequence data.

The working draft of the human genome, completed in February 2001 (Lander et al. 2001), generated virtual reference sequences for each human chromosome, ranging in size from 46 to 246 Mb. NCBI created the first version of its human Map Viewer (Wheeler et al. 2001) shortly thereafter, in order to display these longer sequences. Around the same time, the University of California, Santa Cruz (UCSC) Genome Bioinformatics Group was developing its own human genome browser, based on software originally designed for displaying the much smaller *Caenorhabditis elegans* genome (Kent and Zahler 2000). Similarly, the Ensembl project at the European Molecular Biology Laboratory’s European Bioinformatics Institute (EMBL-EBI) was also producing a system to automatically annotate the human genome sequence, as well as store and visualize the data (Hubbard et al. 2002). The three genome browsers all came online at about the same time, and researchers began using them to help navigate the human genome (Wolfsberg et al. 2002). Today, each site provides free access not only to human sequence data but also to a myriad of other assembled genomic sequences, from commonly used model organisms such as mouse to more recently released assemblies such as those of the domesticated turkey. Although the NCBI’s Map Viewer is not being further developed and will be replaced by its new Genome Data Viewer (Sayers et al. 2019), the UCSC and Ensembl Genome Browsers continue to be popular resources, used by most members of the bioinformatics and genomics communities. This chapter will focus on the last two genome browsers.

The reference human genome was sequenced in a clone-by-clone shotgun sequencing strategy and was declared complete in April 2003, although sequencing of selected regions is still continuing. This strategy includes constructing a bacterial artificial chromosome (BAC) tiling map for each human chromosome, then sequencing each BAC using a shotgun sequencing

approach (reviewed in Green 2001). The sequences of individual BACs were deposited into the High Throughput Genomic (HTG) division of GenBank as they became available. UCSC began assembling these BAC sequences into longer contigs in May 2000 (Kent and Haussler 2001), followed by assembly efforts undertaken at NCBI (Kitts 2003). These contigs, which contained gaps and regions of uncertain order, became the basis for the development of the genome browsers. Over time, as the genome sequence was finished, the human genome assembly was updated every few months. After UCSC stopped producing its own human genome assemblies in August 2001, NCBI built eight reference human genome assemblies for the bioinformatics community, culminating with a final assembly in March 2006. Subsequently, an international collaboration that includes the Wellcome Trust Sanger Institute (WTSI), the Genome Institute at Washington University, EBI, and NCBI formed the Genome Reference Consortium (GRC), which took over responsibility for subsequent assemblies of the human genome. This consortium has produced two human genome assemblies, namely GRCh37 in February 2009 and GRCh38 in December 2013. As one might expect, each new genome assembly leads to changes in the sequence coordinates of annotated features. In between the release of major assemblies, GRC creates patches, which either correct errors in the assembly or add alternate loci. These alternate loci are multiple representations of regions that are too variable to be represented by a single reference sequence, such as the killer cell immunoglobulin-like receptor (KIR) gene cluster on chromosome 19 and the major histocompatibility complex (MHC) locus on chromosome 6. Unlike new genome assemblies, patches do not affect the chromosomal coordinates of annotated features. GRCh38.p10 has 282 alternate loci or patches.

While the GRC also assembles the mouse, zebrafish, and chicken genomes, other genomes are sequenced and assembled by specialized sequencing consortia. The panda genome sequence, published in 2009, was the first mammalian genome to abandon the clone-based sequencing strategies used for human and mouse, relying entirely on next generation sequencing methodologies (Li et al. 2010). Subsequent advances in sequencing technologies have led to rapid increases in the number of complete genome sequences. At the time of this writing, both the UCSC Genome Browser and the main Ensembl web site host genome assemblies of over 100 organisms. The look and feel of each genome browser is the same regardless of the species displayed; however, the types of annotation differ depending on what data are available for each organism.

The backbone of each browser is an assembled genomic sequence. Although the underlying genomic sequence is, with a few exceptions, the same in both genome browsers, each team calculates its annotations independently. Depending on the type of analysis, a user may find that one genome browser has more relevant information than the other. The location of genes, both known and predicted, is a central focus of both genome browsers. For human, at present, both browsers feature the GENCODE gene predictions, an effort that is aimed at providing robust evidence-based reference gene sets (Harrow et al. 2012). Other types of genomic data are also mapped to the genome assembly, including NCBI reference sequences, single-nucleotide polymorphisms (SNPs) and other variants, gene regulatory regions, and gene expression data, as well as homologous sequences from other organisms. Both genome browsers can be accessed through a web interface that allows users to navigate through a graphical view of the genome. However, for those wishing to carry out their own calculations, sequences and annotations can also be retrieved in text format. Each browser also provides a sequence search tool – BLAT (Kent 2002) or BLAST (Camacho et al. 2009) – for interrogating the data via a nucleotide or protein sequence query. (Additional information on both BLAT and BLAST is provided in Chapter 3.)

In order to provide stability and ensure that old analyses can be reproduced, both genome browsers make available not only the current version of the genome assemblies but older ones as well. In addition, annotation tracks, such as the *GENCODE* gene track and the *SNP* track, may be based on different versions of the underlying data. Thus, users are encouraged to verify the version of all data (both genome assembly and annotations) when comparing a region of interest between the UCSC and Ensembl Genome Browsers.

This chapter presents general guidelines for accessing the genome sequence and annotations using the UCSC and Ensembl Genome Browsers. Although similar analyses could be carried out with either browser, we have chosen to use different examples at the two sites to illustrate different types of questions that a researcher might want to ask. We finish with a short description of JBrowse (Buels et al. 2016), another web-based genome browser that users can set up on their own servers to share custom genome assemblies and annotations. All of the resources discussed in this chapter are freely available.

The UCSC Genome Browser

After starting in 2000 with just a display of an early draft of the human genome assembly, the UCSC Genome Browser now provides access to assemblies and annotations from over 100 organisms (Haeussler et al. 2019). The majority of assemblies are of mammalian genomes, but other vertebrates, insects, nematodes, deuterostomes, and the Ebola virus are also included. The assemblies from some organisms, including human and mouse, are available in multiple versions. New organisms and assembly versions are added regularly.

The UCSC Browser presents genomic annotation in the form of tracks. Each track provides a different type of feature, from genes to SNPs to predicted gene regulatory regions to expression data. Each organism has its own set of tracks, some created by the UCSC Genome Bioinformatics team and others provided by members of the bioinformatics community. Over 200 tracks are available for the GRCh37 version of the human genome assembly. The newer human genome assembly, GRCh38, has fewer tracks, as not all the data have been remapped from the older assembly. Other genomes are not as well annotated as human; for example, fewer than 20 tracks are available for the sea hare. Some tracks, such as those created from NCBI transcript data, are updated weekly, while others, such as the SNP tracks created from NCBI variant data (Sayers et al. 2019), are updated less frequently, depending on the release schedule of the underlying data. For ease of use, tracks are organized into subsections. For example, depending on the organism, the *Genes and Gene Predictions* section may include evidence-based gene predictions, ab initio gene predictions, and/or alignment of protein sequences from other species.

The home page of the UCSC Genome Browser provides a stepping-off point for many of the resources developed by the Genome Bioinformatics group at UCSC, including the Genome Browser, BLAT, and the Table Browser, which will be described in detail later in this chapter. The *Tools* menu provides a link to *liftOver*, a widely used tool that converts genomic coordinates from one assembly to another. Using this tool, it is possible to update annotation files so that old data can be integrated into a new genome assembly. The *Download* menu provides an option to download all the sequence and annotation data for each genome assembly hosted by UCSC, as well as some of the source code. The *What's New* section provides updates on new genome assemblies, as well as new tools and features. Finally, there is an extensive *Help* menu, with detailed documentation as well as videos. Users may also submit questions to a mailing list, and most queries are answered within a day.

The UCSC Genome Browser provides multiple ways for both individual users and larger genome centers to share data with collaborators or even the entire bioinformatics community. These sharing options are available on the *My Data* link on the home page. *Custom Tracks* allow users to display their own data as a separate annotation track in the browser. User data must be formatted in a standard data structure in order to be interpreted correctly by the browser. Many commonly used file formats are supported, including Browser Extensible Data (BED), Binary Alignment/Map (BAM), and Variant Call Format (VCF; Box 4.1). Small data files can be uploaded or pasted into the Genome Browser for personal use. Larger files must be saved on the user's web server and accessed by URL through the Genome Browser. As anyone with the URL can access the data, this method can be used to share data with collaborators. Alternatively, *Custom Tracks*, along with track configurations and settings, can be shared with selected collaborators using a named *Session*. Some groups choose to make their

Sessions available to the world at large in *My Data* → *Public Sessions*. Finally, groups with very large datasets can host their data in the form of a *Track Hub* so that it can be viewed on the UCSC Genome Browser. When a *Track Hub* is paired with an *Assembly Hub*, it can be used to create a browser for a genome assembly not already hosted by UCSC.

Box 4.1 Common File Types for Genomic Data

Both the UCSC and Ensembl Genome Browsers allow users to upload their own data so that they can be viewed in context with other genome-scale data. User data must be formatted in a commonly used data structure in order to be interpreted correctly by the browser.

Browser Extensible Data (BED) format is a tab-delimited format that is flexible enough to display many types of data. It can be used to display fairly simple features like the location of transcription binding factor sites, as well more complex ones like transcripts and their exons.

Binary Alignment/Map (BAM) format is the compressed binary version of the *Sequence Alignment/Map (SAM) format*. It is a compact format designed for use with very large files of nucleotide sequence alignments. Because it can be indexed, only the portion of the file that is needed for display is transferred to the browser. Many tools for next generation sequence analysis use BAM format as output or input.

Variant Call Format (VCF) is a flexible format for large files of variation data including single-nucleotide variants, insertions/deletions, copy number variants, and structural variants. Like BAM format, it is compressed and indexed, and only the portion of the file that is needed for display is transferred to the browser. Many tools for variant analysis use VCF format as output or input.

The UCSC Genome Browser home page lists commonly accessed tools, as well as a frequently updated news section that highlights major data and software updates. To reach the Genome Browser Gateway, the main entry point for text-based searches, click on the *Gateway* link on the home page (Figure 4.1). The default assembly is the most recent human assembly, GRCh38, from December 2013. The genomes of other species can be selected from the phylogenetic tree on the left side of the *Gateway* page, or by typing their name in the selection box. On the human *Gateway* page, there is also the option to select one of four older human genome assemblies. Details about the GRCh38 assembly and instructions for searching are available on the *Gateway* page.

To perform a search, enter text into the *Position/Search Term* box. If the query maps to a unique position in the genome, such as a search for a particular chromosome and position, the *Go* button links directly to the Genome Browser. However, if there is more than one hit for the query, such as a search for the term *metalloprotease*, the resulting page will contain a list of results that all contain that term. For some species, the terms have been indexed, and typing a gene symbol into the search box will bring up a list of possible matches. In this example, we will search for the human hypoxia inducible factor 1 alpha subunit (*HIF1A*) gene (Figure 4.1), which produces a single hit on GRCh38.

The default Genome Browser view showing the genomic context of the *HIF1A* gene is shown in Figure 4.2. The navigation controls are presented across the top of the display. The arrows move the window to the left and right along the chromosome. Alternatively, the user can move the display left and right by holding down the mouse button and dragging the window. To zoom in and out, use the buttons at the top of the display. The *base* button zooms in so far that individual nucleotides are displayed, while the zoom out 100× button will show the entire chromosome if it is pressed a few times. The current genomic position and the length of window (in nucleotides) is shown above a schematic of chromosome 14, where the current

The screenshot shows the UCSC Genome Browser Gateway interface. The top navigation bar includes links for Genomes, Genome Browser, Tools, Mirrors, Downloads, My Data, Help, and About Us. The main content area is split into two columns. The left column, titled 'Browse/Select Species', features 'POPULAR SPECIES' (Human, Mouse, Rat, Fruitfly, Worm, Yeast) and a 'REPRESENTED SPECIES' phylogenetic tree. A red box highlights the 'Human' species in the tree. The right column, titled 'Find Position', shows a search for 'hif1a' on the 'Human Assembly' (Dec. 2013 GRCh38/hg38). The search results list several HIF1A-related genes and their RefSeq IDs. Below the search results, there are sections for 'UCSC Genome Browser assembly ID', 'Sequencing/Assembly provider ID', 'Assembly date', 'Accession ID', 'NCBI Genome ID', 'NCBI Assembly ID', and 'BioProject ID'. A 'Search the assembly:' section provides instructions on how to use the search box. At the bottom, there is a 'Download sequence and annotation data:' section with options for 'Using rsync (recommended)' and 'Using FTP'.

Figure 4.1 The home page of the UCSC Genome Browser, showing a query for the gene *HIF1A* on the human GRCh38 genome assembly. The organism can be selected by clicking on its name in the phylogenetic tree. For many organisms, more than one genome assembly is available. Typing a term into the *Position/Search Term* box returns a list of matching gene symbols.

genomic position is highlighted with a red box. A new search term can be entered into the search box.

Below the browser window illustrated in Figure 4.2, one would find a list of tracks that are available for display on the assembly. The tracks are separated into nine categories: Mapping and Sequencing, Genes and Gene Predictions, Phenotype and Literature, mRNA and Expressed Sequence Tag (EST), Expression, Regulation, Comparative Genomics, Variation, and Repeats. Clicking on a track name opens the *Track Settings* page for that track, providing a description of the data displayed in that track. Most tracks can be displayed in one of the following five modes.

- 1) *Hide*: the track is not displayed at all.
- 2) *Dense*: all features are collapsed into a single line; features are not labeled.
- 3) *Squish*: each feature is shown separately, but at 50% the height of *full* mode; features are not labeled.
- 4) *Pack*: each feature is shown separately, but not necessarily on separate lines; features are labeled.
- 5) *Full*: each feature is labeled and displayed on a separate line.

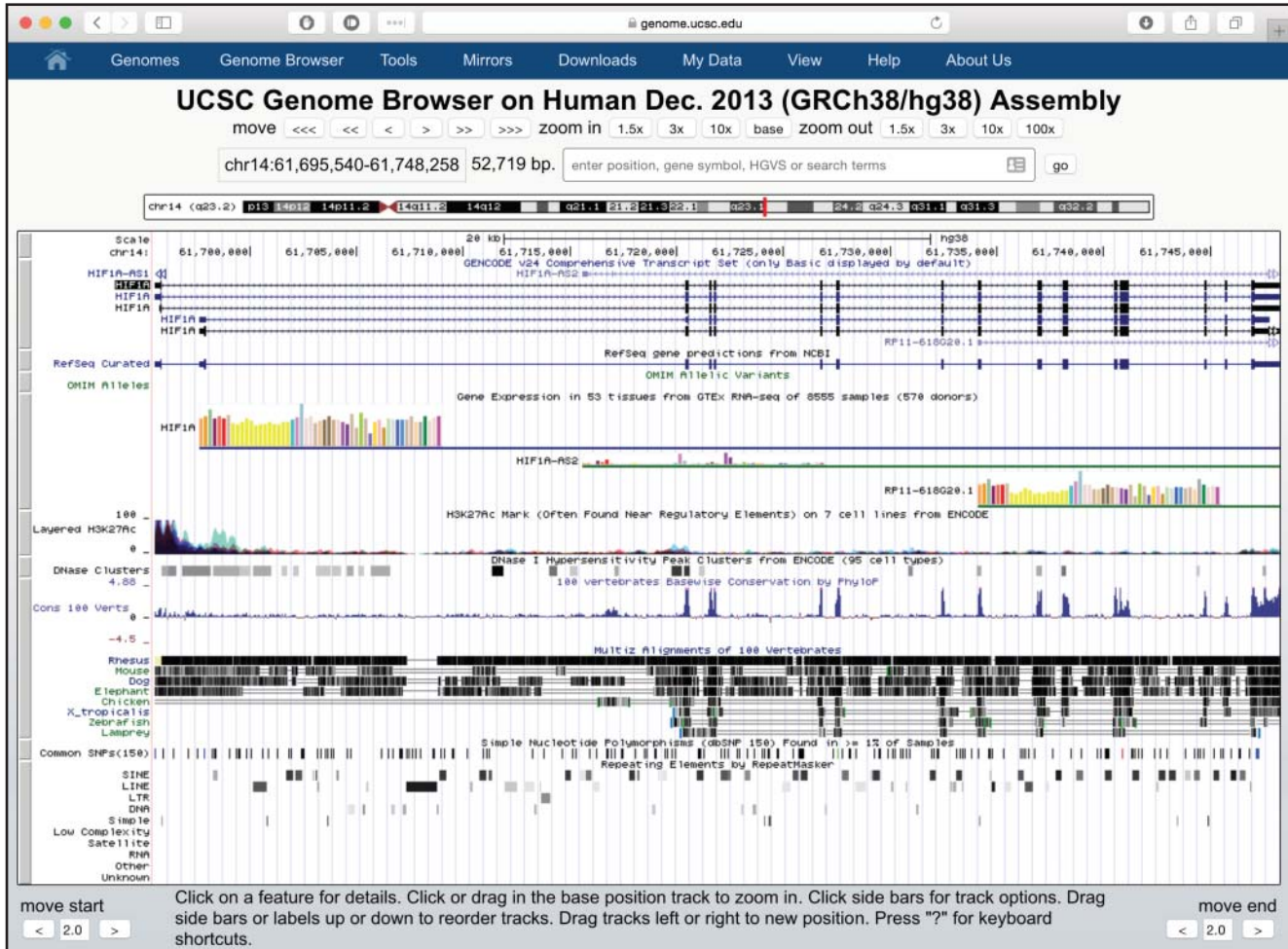


Figure 4.2 The default view of the UCSC Genome Browser, showing the genomic context of the human *HIF1A* gene.

In order to simplify the display, most tracks are in *hide* mode by default. To change the mode, use the pull-down menu below the track name or on the *Track Settings* page. Other settings, such as color or annotation details, can also be configured on the *Track Settings* page. For example, the *NCBI RefSeq* track allows users to select if they want to view all reference sequences or only those that are curated or predicted (Box 1.2). One possible point of confusion is that the UCSC Genome Browser will “remember” the mode in which each track is displayed from session to session. Custom settings can be cleared by selecting *Reset all User Settings* under the *Genome Browser* pull-down menu at the top of any page.

The annotation tracks in the window below the chromosome are the focus of the Genome Browser (Figure 4.2). Tracks are depicted horizontally, with a title above the track and labels on the left. The first two lines show the scale and chromosomal position. The term that was searched for and matched (*HIF1A* in this case) is highlighted on the annotation tracks. The next tracks shown by default are gene prediction tracks. The default gene track on GRCh38 is the GENCODE Genes set, which replaces the UCSC Genes track that is still displayed on GRCh37 and older human assemblies. GENCODE genes are annotated using a combination of computational analysis and manual curation, and are used by the ENCODE Consortium and other groups as reference gene sets (Box 4.2). The *GENCODE v24* track depicts all of the gene models from the GENCODE v24 release, which includes both protein-coding genes and non-coding RNA genes.

Box 4.2 GENCODE

The GENCODE gene set was originally developed by the ENCODE Consortium as a comprehensive source of high-quality human gene annotations (Harrow et al. 2012). It has now been expanded to include the mouse genome (Mudge and Harrow 2015). The goal of the GENCODE project is to include all alternative splice variants of protein-coding loci, as well as non-coding loci and pseudogenes. The GENCODE Consortium uses computational methods, manual curation, and experimental validation to identify these gene features. The first step is carried out by the same Ensembl gene annotation pipeline that is used to annotate all vertebrate genomes displayed at Ensembl (Aken et al. 2016). This pipeline aligns cDNAs, proteins, and RNA-seq data to the human genome in order to create candidate transcript models. All Ensembl transcript models are supported by experimental evidence; no models are created solely from ab initio predictions. The Human and Vertebrate Analysis and Annotation (HAVANA) group produces manually curated gene sets for several vertebrate genomes, including mouse and human. These manually curated genes are merged with the Ensembl transcript models to create the GENCODE gene sets for mouse and human. A subset of the human models has been confirmed by an experimental validation pipeline (Howald et al. 2012).

The consortium makes available two types of GENCODE gene sets. The Comprehensive set encompasses all gene models, and may include many alternatively spliced transcripts (isoforms) for each gene. The Basic set includes a subset of representative transcripts for each gene that prioritizes full-length protein-coding transcripts over partial- or non-protein-coding transcripts. The Ensembl Genome Browser displays the Comprehensive set by default. Although the UCSC Genome Browser displays the Basic set by default, the Comprehensive set can be selected by changing the *GENCODE* track settings. At the time of this writing, Ensembl is displaying GENCODE v27, released in August 2017. The GENCODE version available by default at the UCSC Genome Browser is v24, from December 2015. More recent versions of GENCODE can be added to the browser by selecting them in the *All GENCODE* super-track.

GENCODE and RefSeq both aim to provide a comprehensive gene set for mouse and human. Frankish et al. (2015) have shown that, in human, the RefSeq gene set is more similar to the GENCODE Basic set, while the GENCODE Comprehensive set contains more alternative splicing and exons, as well as more novel protein-coding sequences, thus covering more of the genome. They also sought to determine which gene set would provide the best reference transcriptome for annotating variants. They found that the GENCODE Comprehensive set, because of its better genomic coverage, was better for discovering new variants with functional potential, while the GENCODE Basic set may be better suited for applications where a less complex set of transcripts is needed. Similarly, Wu et al. (2013) compared the use of different gene sets to quantify RNA-seq reads and determine gene expression levels. Like Frankish et al., they recommend using less complex gene annotations (such as the RefSeq gene set) for gene expression estimates, but more complex gene annotations (such as GENCODE) for exploratory research on novel transcriptional or regulatory mechanisms.

In the *GENCODE* track, as well as other gene tracks, exons (regions of the transcript that align with the genome) are depicted as blocks, while introns are drawn as the horizontal lines that connect the exons. The direction of transcription is indicated by arrowheads on the introns. Coding regions of exons are depicted as tall blocks, while non-coding exons are shorter. In this example, the *GENCODE* track depicts five alternatively spliced transcripts, labeled *HIF1A* on the left, for the *HIF1A* gene. As shown by the arrowheads, all transcripts are transcribed from left to right. The 5'-most exon of each transcript (on the left side of the display) is shorter on the left, indicating an untranslated region (UTR), and

(Continued)

Box 4.2 (Continued)

taller on the right, indicating a coding sequence. The reverse is true for the 3'-most exon of each transcript. A very close visual inspection of the Genome Browser shows that the last four *HIF1A* transcripts have a different pattern of exons from each other; a BLAST search (not shown) reveals that first two transcripts differ by only three nucleotides in one exon. There is also a transcript labeled *HIF1A-AS2*, an anti-sense *HIF1A* transcript that is transcribed from right to left. Another transcript, labeled *RP11-618G20.1*, is a synthetic construct DNA. Zooming the display out by 3× allows a view of the genes immediately upstream and downstream of *HIF1A* (Figure 4.3). A second *HIF1A* antisense transcript, *HIF1A-AS1*, is also visible.

The track below the *GENCODE* track is the *RefSeq gene predictions from NCBI* track. This is a composite track showing human protein-coding and non-protein-coding genes taken from the NCBI RNA reference sequences collection (RefSeq; Box 1.2). By default, the *RefSeq* track is shown in *dense* mode, with the exons of the individual transcripts condensed into a single line (Figure 4.2). Note that, in this *dense* mode, the exons are displayed as blocks, as in the *GENCODE* track, but there are no arrowheads on the gene model to show the direction of transcription. To change the display of the *RefSeq* track to view individual transcripts, open the *Track Settings* page for the *NCBI RefSeq* track by clicking on the track name in the first row

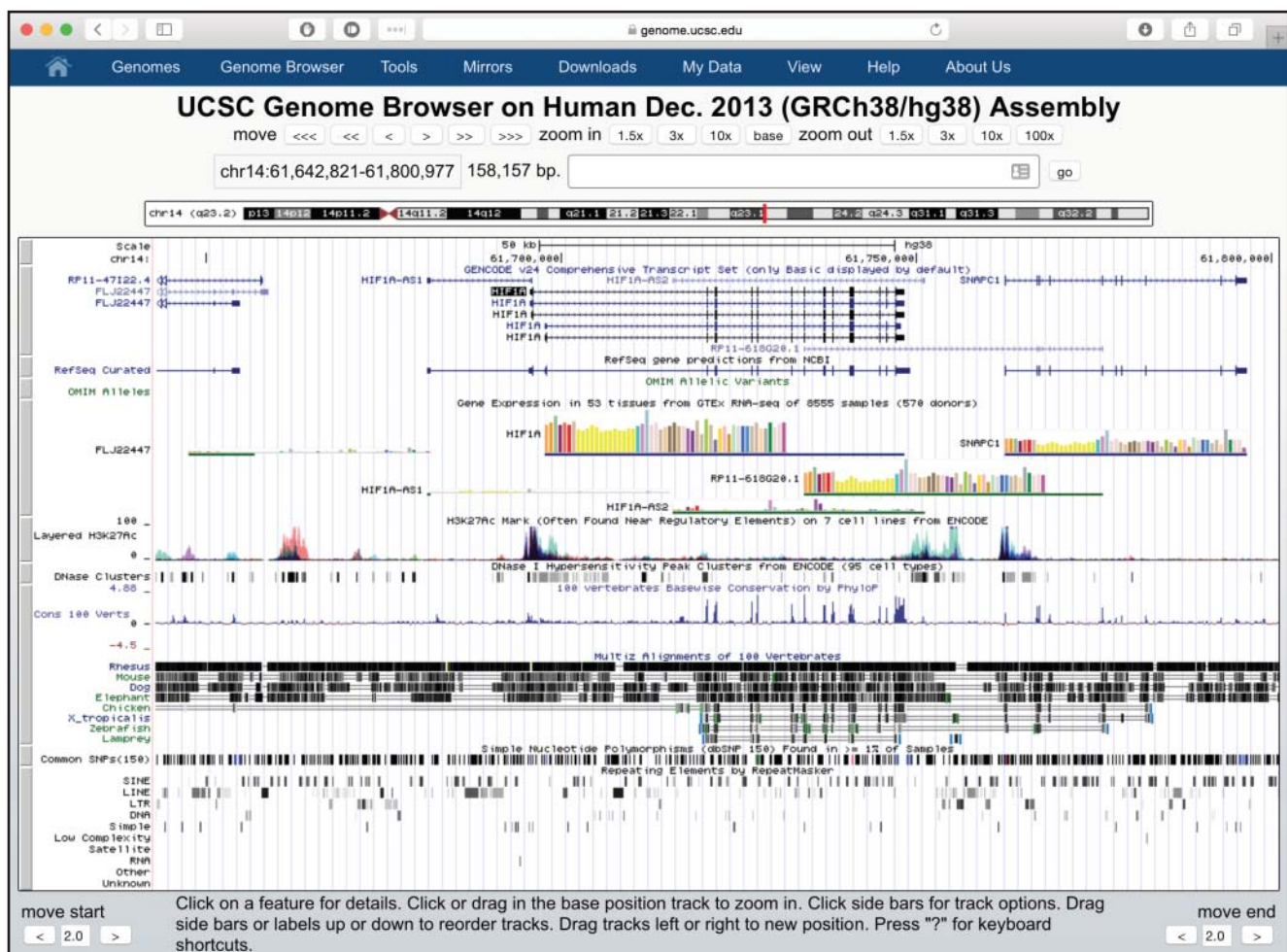


Figure 4.3 The genomic context of the human *HIF1A* gene, after clicking on *zoom out* 3×. The genes immediately upstream (*FLJ22447*) and downstream (*SNAPC1*) of *HIF1A* are now visible.

Figure 4.4 The *RefSeq Track Settings* page. The track settings pages are used to configure the display of annotation tracks. By default, all of the *RefSeq* tracks are set to display in *dense* mode, with all features condensed into a single line. In this example, the Curated RefSeqs are being set to display in *full* mode, in which each RefSeq transcript will be labeled and displayed on a separate line. The remainder of the RefSeqs will be displayed in *dense* mode. The types of RefSeqs, curated and predicted, are described in Box 1.2. After changing the settings, press the *submit* button to apply them.

of the *Genes and Gene Predictions* section (below the graphical view shown in Figure 4.2). The resulting *Track Settings* page (Figure 4.4) allows the user to choose which type of RefSeqs to display (e.g. all, curated only, or predicted only). In this example, we change the mode of the *RefSeq Curated* track from *dense* to *full*, and the resulting graphical view (Figure 4.5) displays each curated RefSeq as a separate transcript. In contrast to the *GENCODE* track, there are only three RefSeq transcripts for the *HIF1A* gene, and the *HIF1A-AS2* RefSeq transcript is much shorter than the *GENCODE* transcript with the same name. These discrepancies are due to differences in how the RefSeq and *GENCODE* transcript sets are assembled (Boxes 1.2 and 4.2).

Additional information about each transcript in the *GENCODE* and *RefSeq* tracks is available by clicking on the gene symbol (*HIF1A*, in this case); as the original search was for *HIF1A*,

Figure 4.5 The genomic context of the human *HIF1A* gene, after displaying RefSeq Curated genes in *full* mode. Each RefSeq transcript is now drawn on a separate line, so that individual exons, as well as the direction of transcription, are visible. Compare this rendition with Figure 4.2, where all RefSeq transcripts are condensed on a single line.

Genomic Sequence Near Gene

Get Genomic Sequence Near Gene

Note: if you would prefer to get DNA for more than one feature of this track at a time, try the [Table Browser](#) using the output format sequence.

Sequence Retrieval Region Options:

- Promoter/Upstream by bases
- 5' UTR Exons
- CDS Exons
- 3' UTR Exons
- Introns
- Downstream by bases
- One FASTA record per gene.
- One FASTA record per region (exon, intron, etc.) with extra bases upstream (5') and extra downstream (3')
 - Split UTR and CDS parts of an exon into separate FASTA records

Note: if a feature is close to the beginning or end of a chromosome and upstream/downstream bases are added, they may be truncated in order to avoid extending past the edge of the chromosome.

Sequence Formatting Options:

- Exons in upper case, everything else in lower case.
- CDS in upper case, UTR in lower case.
- All upper case.
- All lower case.
- Mask repeats: to lower case to N

```
>hg38_knownGene_uc001xfq.3 range=chr14:61694540-61695539 5'pad=0 3'pad=0 strand=+ repeatMasking=none
ttagaagactgacaggcttgaagttatgctccaagaacaaaagtgatt
atgtttttagtagcttgcacaaagtgccaaagccatttttctactc
ttccctgaaattggttatagcttattaagtcattatacctatttg
caaatgcttaacatagtttcagatttaagattccctgcaactttatt
ccctgaagtttacagcaacaggagtccatttttattttaattgcatt
attcagttaactccgcccacagaaaaacttagtagacaaggtgagtt
ccctgtgctccgtggcaaaagagtgcggtgggtgacattgacctggtt
agtaactctgtaagaaagaccctgtgtaacacatctgagcaacgaga
ccaaaggaagggcttgcgccacgaggcgaagtctgctttttgaacaga
gagcccagcagagttggcggaatcgtgccagcactgaggccgaggag
aaagagagcaggagcattacattactgcccagagtaggaaaatgat
gcatgttgggaccaggcaaccgaaatcccttctcagcagcgcctccaa
agccgggacaccgcttccctcgagagaagggcagagtcaccagactcggg
ctgagccgcacccccatctcctttctctttctccgcccgtaaacacaga
cgagcagctgagcgtcgcagcccgtcccagctgtgctcagctgaccgcc
tctgatggctgagagcggcgtgggtgggtggggaacttgccgctgc
gtcgtcgcattggatctcgaggaaccccctccacctcaggtgagggc
ggcttgcgggagcgcgcgcgctgggagggcagcggggcgcctccg
ccccctccccctccccgcgcgcgcgagcgcgcctccgcccctgcccgc
cctgacgctgctcagctcctcagtgcaagtgctgctcgtctgaggg
```

Figure 4.6 The *Get Genomic Sequence* page that provides an interface for users to retrieve the sequence for a feature of interest. Click on an individual transcript in the *GENCODE* or *RefSeq* track to open a page with additional details for that transcript. On either of those details pages, click the link for *Genomic Sequence* to open the page displayed here, which provides choices for retrieving sequences upstream or downstream of the transcript, as well as intron or exon sequences. In this example, retrieve the sequence 1000 nt upstream of the annotated transcription start site. Shown in the inset is the result of retrieving the FASTA-formatted sequence 1000 nt upstream of the *HIF1A* transcript.

the gene name is highlighted in inverse type. For *GENCODE* genes, UCSC has collected information from a variety of public sources and includes a text description, accession numbers, expression data, protein structure, Gene Ontology terms, and more. For *RefSeq* transcripts, UCSC provides links to NCBI resources. Both *GENCODE* and *RefSeq details* pages provide a link to *Genomic Sequence* in the *Sequence and Links* section, allowing users to retrieve genomic sequences connected to an individual transcript. From the selection menu (Figure 4.6), users can choose whether to download the sequence upstream or downstream of the gene, as well as the exon or intron sequence. The sequence is returned in FASTA format.

Further down on the graphical view shown in Figure 4.3 are tracks from the *ENCODE Regulation* super-track: *Layered H3K27Ac* and *DNase Clusters*. These data were generated by the Encyclopedia of DNA Elements (ENCODE) Consortium between 2003 and 2012 (ENCODE Project Consortium 2012). The ENCODE Consortium has developed reagents and tools to identify all functional elements in the human genome sequence. The *Layered H3K27Ac* track indicates regions where there are modified histones that may indicate active enhancers (Box 4.3).

Box 4.3 Histone Marks

Histone proteins package DNA into chromosomes. Post-translational modifications of these histones can affect gene expression, as well as DNA replication and repair, by changing chromatin structure or recruiting histone modifiers (Lawrence et al. 2016). The post-translational modifications include methylation, phosphorylation, acetylation, ubiquitylation, and sumoylation. Histone H3 is primarily acetylated on lysine residues, methylated at arginine or lysine, or phosphorylated on serine or threonine. Histone H4 is primarily acetylated on lysine, methylated at arginine or lysine, or phosphorylated on serine.

Histone modification (or “marking”) is identified by the name of the histone, the residue on which it is marked, and the type of mark. Thus, H3K27Ac is histone H3 that is acetylated on lysine 27, while H3K79me2 is histone H3 that is dimethylated on lysine 79. Different histone marks are associated with different types of chromatin structure. Some are more likely found near enhancers and others near promoters and, while some cause an increase of expression from nearby genes, others cause less. For example, H3K4me3 is associated with active promoters, and H3K27me3 is associated with developmentally controlled repressive chromatin states.

The *DNase Clusters* track depicts regions where chromatin is hypersensitive to cutting by the DNaseI enzyme. In these hypersensitive regions, the nucleosome structure is less compacted, meaning that the DNA is available to bind transcription factors. Thus, regulatory regions, especially promoters, tend to be DNase sensitive. The track settings for the *ENCODE Regulation* super-track allows other ENCODE tracks to be added to the browser window, including additional histone modification and DNaseI hypersensitivity data. Changing the display of the *H3K4Me3* peaks from *hide* to *full* highlights the peaks in the *H3K4Me3* track near the 5' ends of the *HIF1A* and *SNAPC1* transcripts that overlap with DNase hypersensitive sites (Figure 4.7, blue highlights). These peaks may represent promoter elements that regulate the start of transcription.

The UCSC Genome Browser displays data from NCBI’s Single Nucleotide Polymorphism Database (dbSNP) in four SNP tracks. *Common SNPs* contains SNPs and small insertions and deletions (indels) from NCBI’s dbSNP that have a minor allele frequency of at least 1% and are mapped to a single location in the genome. Researchers looking for disease-causing SNPs can use this track to filter their data, hypothesizing that their variant of interest will be rare and therefore not displayed in this track. *Flagged SNPs* are those that are deemed by NCBI to be clinically associated, while *Mult. SNPs* have been mapped to more than one region in the genome. NCBI filters out most multiple-mapping SNPs as they may not be true SNPs, so there are not many variants in this track. *All SNPs* includes all SNPs from the three subcategories. dbSNP is in a continuous state of growth, and new data are incorporated a few times each year as a new release, or new build, of dbSNP. These four SNP tracks are available for a few of the most recent builds of dbSNP, indicated by the number in the track name. Thus, for example, *Common SNPs (150)* are SNPs found in $\geq 1\%$ of samples from dbSNP build 150.

By default, the *Common SNPs (150)* track is displayed in *dense* mode, with all variants in the region compressed onto a single line. Variants in the *Common SNPs* track are color coded by function. Open the *Track Settings* for this track in order to modify the display (Figure 4.8). Set the *Display mode* to *pack* in order to show each variant separately. At the same time, modify the *Coloring Options* so that SNPs in UTRs of transcripts are set to blue and SNPs in coding regions of transcripts are set to green if they are synonymous (no change to the protein sequence) or red if they are non-synonymous (altering the protein sequence), with all remaining classes of SNPs set to display in black. Note the changes in the resulting browser window, with the green synonymous and blue untranslated SNPs clearly visible (Figure 4.9).

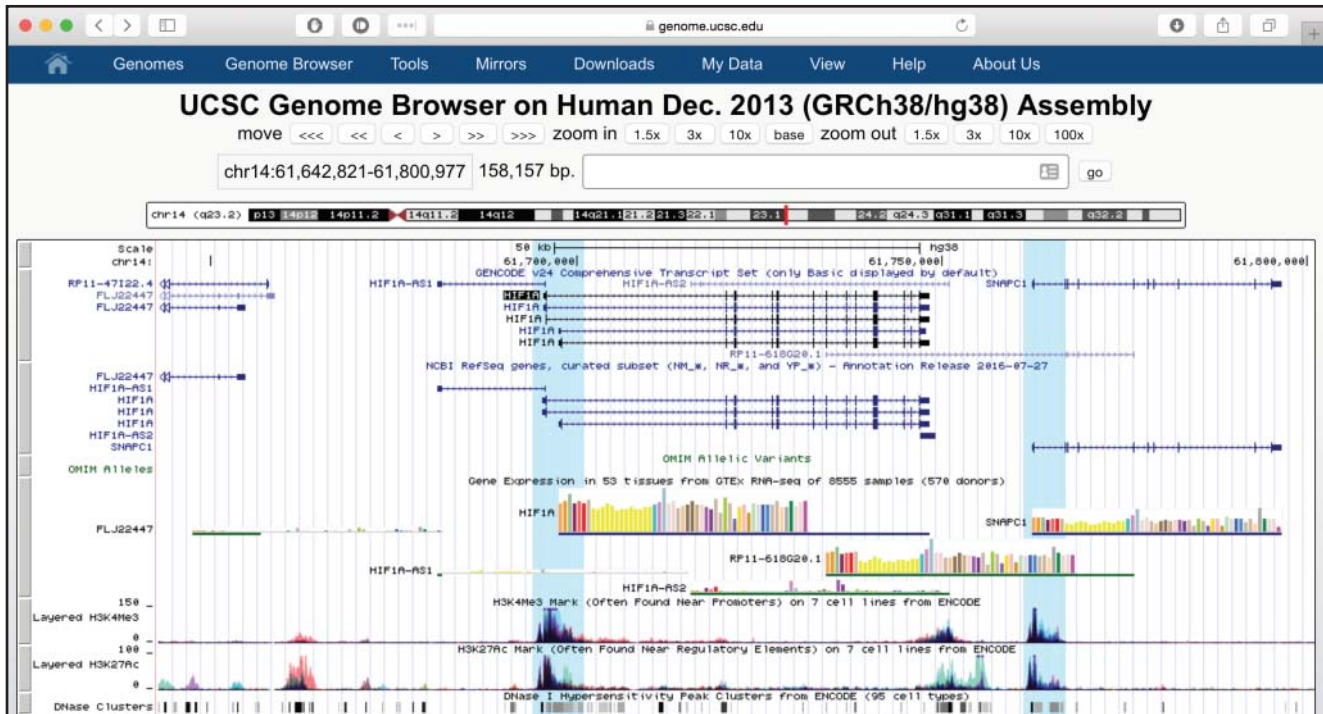


Figure 4.7 The genomic context of the human *HIF1A* gene, after changing the display of the *H3K4Me3* peaks from *hide* to *full*. The *H3K4Me3* track is part of the *ENCODE Regulation Super-track*. Below the graphic display window in Figure 4.5, open up the *ENCODE Regulation Super-track*, in the *Regulation* menu. Change the track display from *hide* to *full* to reproduce the page shown here. Note that the *H3K4Me3* peaks, which can indicate promoter regions (Box 4.3), overlap with the transcription starts of the *SNRPC1* and *HIF1A* genes (light blue highlight). These regions also overlap with the *DNase HS* track, indicating that the chromatin should be available to bind transcription factors in this region. The highlights were added within the Genome Browser using the *Drag-and-select* tool. This tool is accessed by clicking anywhere in the *Scale* track at the top of the Genome Browser display and dragging the selection window across a region of interest. The *Drag-and-select* tool provides options to *Highlight* the selected region or *Zoom* directly to it.

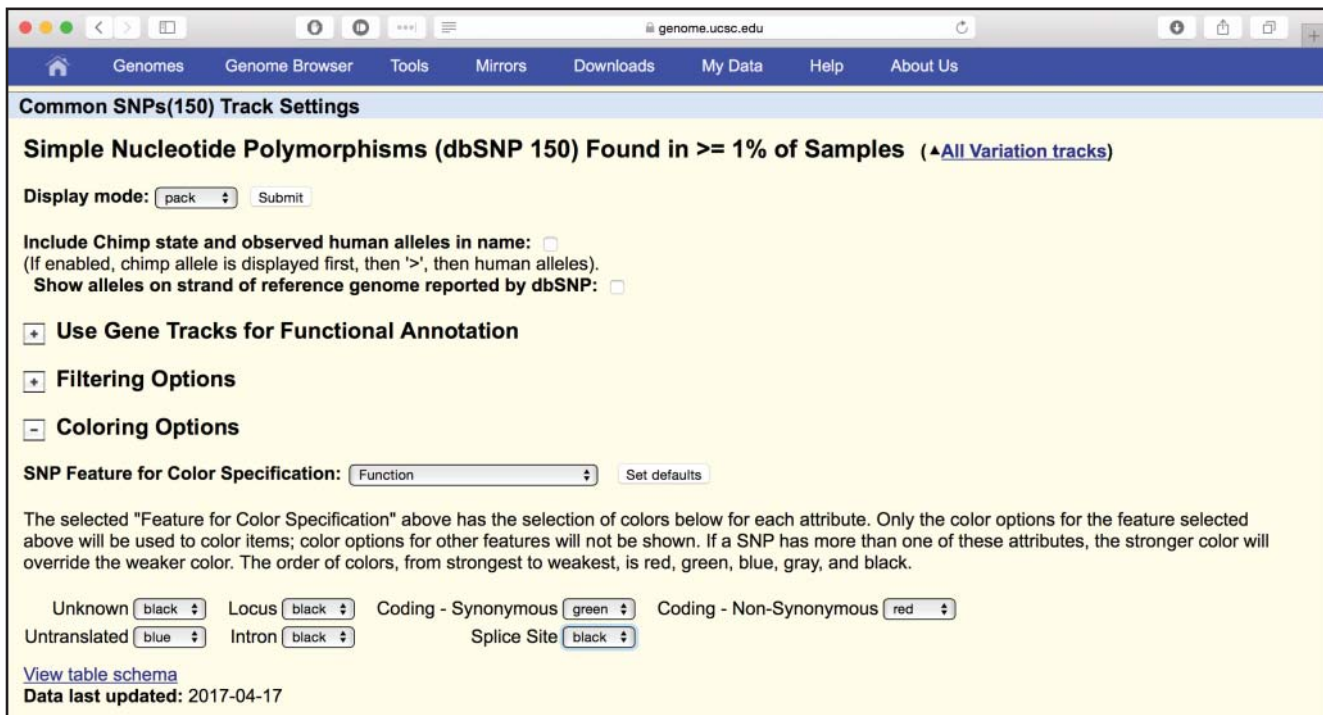


Figure 4.8 Configuring the track settings for the *Common SNPs(150)* track. Set the *Coloring Options* so that all SNPs are black, except for untranslated SNPs (blue), coding-synonymous SNPs (green), and coding-non-synonymous SNPs (red). In addition, change the *Display mode* of the track from *dense* to *pack* so that the individual SNPs can be seen. By default, the position of each variant is defined by its position within transcripts in the *GENCODE* track. However, the track used for annotation can be changed in the settings called *Use Gene Tracks for Functional Annotation*.



Figure 4.9 The genomic context of the human *HIF1A* gene, after changing the colors and display mode of the *Common SNPs(150)* track as shown in Figure 4.8. The SNPs in the 5' and 3' untranslated regions of the *HIF1A* GENCODE transcripts are now colored blue, while the coding-synonymous SNP is colored green.

Two types of *Expression* tracks display data from the NIH Genotype-Tissue Expression (GTEx) project (GTEx Consortium 2015). The *GTEx Gene* track displays gene expression levels in 51 tissues and two cell lines, based on RNA-seq data from 8555 samples. The *GTEx Transcript* track provides additional analysis of the same data and displays median transcript expression levels. By default, the *GTEx Gene* track is shown in *pack* mode, while the *GTEx Transcript* track is hidden. Figure 4.10 shows the *Gene* track in *pack* display mode, in the region of the phenylalanine hydroxylase (*PAH*) gene. The height of each bar in the bar graph represents the median expression level of the gene across all samples for a tissue, and the bar color indicates the tissue. The *PAH* gene is highly expressed in kidney and liver (the two brown bars). The expression is more clearly visible in the details page for the *GTEx* track (Figure 4.10, inset, purple box). The *GTEx Transcript* track is similar, but depicts expression for individual transcripts rather than an average for the gene.

An alternate entry point to the UCSC Genome Browser is via a *BLAT search* (see Chapter 3), where a user can input a nucleotide or protein sequence to find an aligned region in a selected genome. BLAT excels at quickly identify a matching sequence in the same or highly similar organism. We will attempt to use BLAT to find a lizard homolog of the human gene

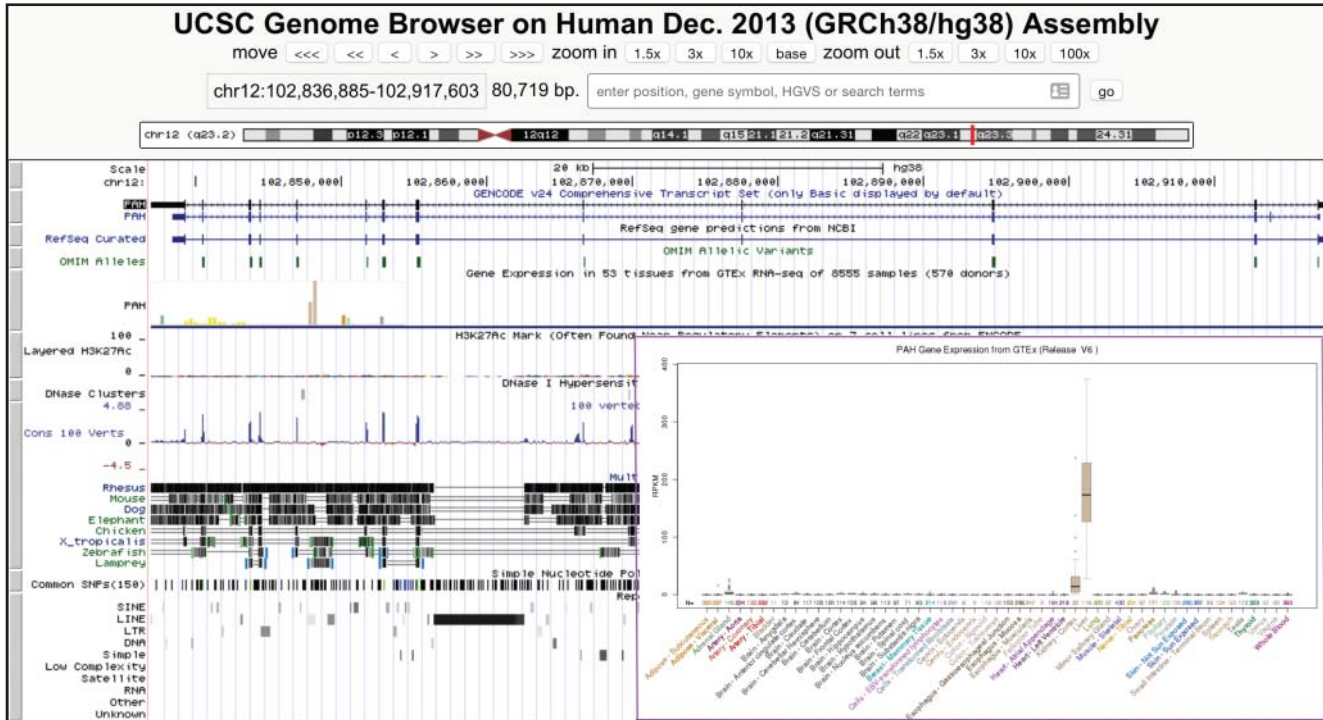


Figure 4.10 The *GTEX Gene* track, which depicts median gene expression levels in 51 tissues and two cell lines, based on RNA-seq data from the *GTEX* project from 8555 tissue samples. The main browser window depicts the *GTEX Gene* track for the human *PAH* gene, showing high expression in the two tissues colored brown (liver and kidney) but low or no expression in others. Clicking on the *GTEX* track opens it in a larger window, shown in the inset.

disintegrin and metalloproteinase domain-containing protein 18 (*ADAM18*). The *ADAM18* protein sequence is copied in FASTA format from the NCBI view of accession number NP_001307242.1 and pasted into the *BLAT Search* box that can be accessed from the *Tools* pull-down menu; the method for retrieving this sequence in the correct format is described in Chapter 2. Select the lizard genome and assembly AnoCar2.0/anoCar2. *BLAT* will automatically determine that the query sequence is a protein and will compare it with the lizard genome translated in all six reading frames. A single result is returned (Figure 4.11a). The alignment between the *ADAM18* protein sequence and lizard chromosome Un_GL343418 runs from amino acid 368 to amino acid 383, with 81.3% identity. The *browser* link depicts the genomic context of this 48 nt hit (Figure 4.11b). Although the *ADAM18* protein sequence aligns to a region in which other human *ADAM* genes have also been aligned, the other human genes are represented by a thin line, indicating a gap in their alignment. The *details* link shown in Figure 4.11a produces the alignment between the *ADAM18* protein and lizard chromosome Un_GL343418 (Figure 4.11c). The top section of the results shows the protein query sequence, with the blue letters indicating the short region of alignment with the genome. The bottom section shows the pairwise alignment between the protein and genomic sequence translated in six frames. Vertical black lines indicate identical sequences. Taken together, the *BLAT* results show that only 16 amino acids of the 715 amino acid *ADAM18* protein align to the lizard genome (Figure 4.11c). This alignment is short and likely does not represent a homologous region between the *ADAM18* protein and the lizard genome. Thus, the *BLAT* algorithm, although fast, is not always sensitive enough to detect cross-species orthologs. The *BLAST* algorithm, described in the Ensembl Genome Browser section, is more sensitive, and is a better choice for identifying such homologs.

Lizard BLAT Results

BLAT Search Results

Go back to [chrUn_GL343418:612590-612637](#) on the Genome Browser.

Custom track name: Custom track description:

ACTIONS	QUERY	SCORE	START	END	QSIZE	IDENTITY	CHRO	STRAND	START	END	SPAN
browser details	NP_001307242.1	30	368	383	715	81.3%	Un_GL343418	++	612590	612637	48

[Missing a match?](#)

(a)

UCSC Genome Browser on Lizard May 2010 (Broad AnoCar2.0/anoCar2) Assembly

move <<< << < > >> >>> zoom in 1.5x 3x 10x base zoom out 1.5x 3x 10x 100x

chrUn_GL343418:612,590-612,637 48 bp.

chrUn_GL343418 | Un_GL343418

Scale | 10 bases | chrUn_GL343418 | anoCar2

chrUn_GL343418: 612,595 | 612,600 | 612,605 | 612,610 | 612,615 | 612,620 | 612,625 | 612,630 | 612,635

Gap Locations

Your Sequence from BLAT Search

NP_001307242.1

Human Proteins Mapped by Chained tBLASTn

Other RefSeq

Spliced ESTs

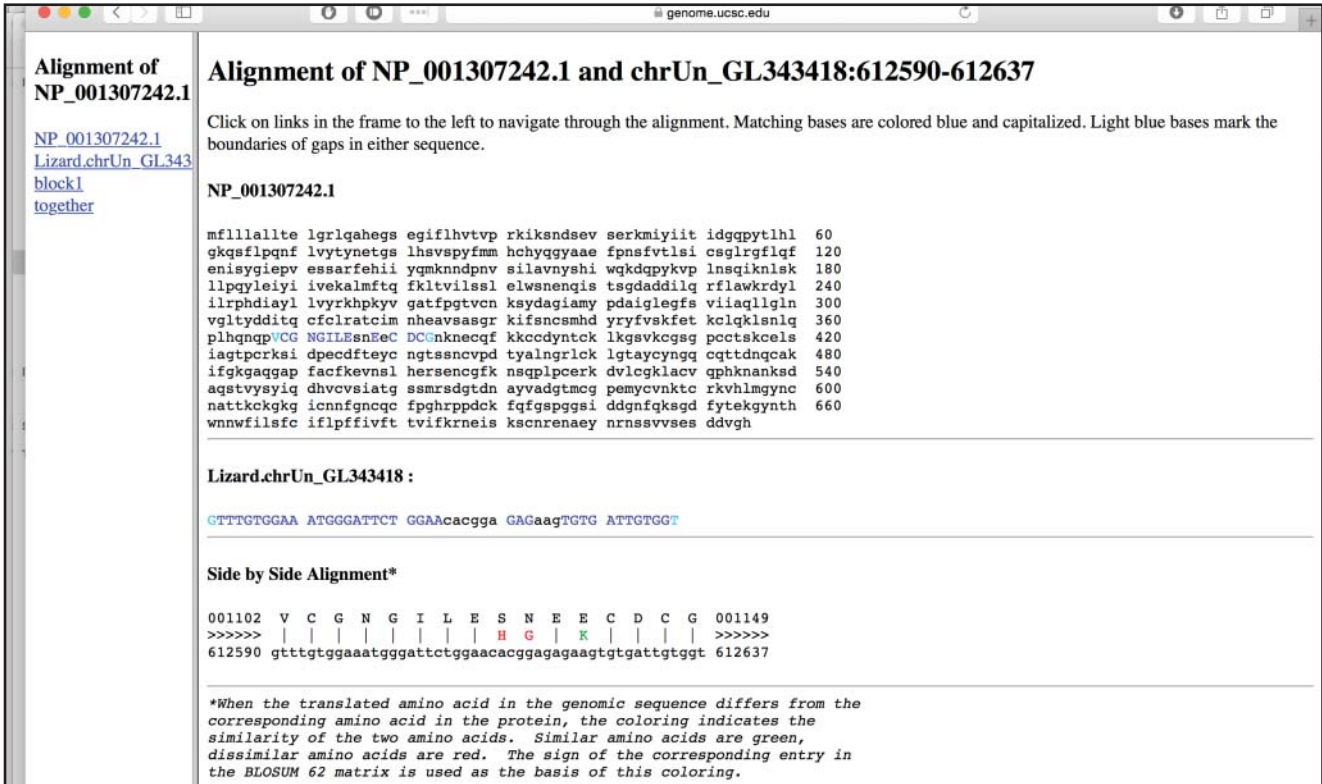
Multiz Alignments & Conservation (7 species)

Lizard | Chicken | Platypus | Human | Mouse | X_tropicalis | Stickleback

RepeatMasker

(b)

Figure 4.11 BLAT search at the UCSC Genome Browser. (a) This page shows the results of running a BLAT search against the lizard genome, using as a query the human protein sequence of the gene *ADAM18*, accession NP_001307242.1. The ADAM18 protein sequence is available from NCBI at www.ncbi.nlm.nih.gov/protein/NP_001307242.1?report=fasta. At the UCSC Genome Browser, the web interface to the BLAT search is in the *Tools* menu at the top of each page. The BLAT search was run against the lizard genome assembly from May 2010, also called anoCar2. The columns on the results page are as follows: **ACTIONS**, links to the browser (Figure 4.11b) and details (Figure 4.11c); **QUERY**, the name of the query sequence; **SCORE**, the BLAT score, determined by the number of matches vs. mismatches in the final alignment of the query to the genome; **START**, the start coordinate of the alignment, on the query sequence; **END**, the end coordinate of the alignment, on the query sequence; **QSIZE**, the length of the query; **IDENTITY**, the percent identity between the query and the genomic sequences; **CHRO**, the chromosome to which the query sequence aligns; **STRAND**, the chromosome strand to which the query sequence aligns; **START**, the start coordinate of the alignment, on the genomic sequence; **END**, the end coordinate of the alignment, on the genomic sequence; and **SPAN**, the length of the alignment, on the genomic sequence. Note that, in this example, there is a single alignment; searches with other sequences may result in many alignments, each shown on a separate line. It is possible to search with up to 25 sequences at a time, but each sequence must be in FASTA format. (b) This page shows the *browser* link from the BLAT summary page. The alignment between the query and genome is shown as a new track called *Your Sequence from BLAT Search*. (c) The *details* link from the BLAT summary page, showing the alignment between the query (human ADAM18 protein) and the lizard genome, translated in six frames. The protein query sequence is shown at the top, with the blue letters indicating the amino acids that align to the genome. The bottom section shows the pairwise alignment between the protein and genomic sequence translated in six frames. Black lines indicate identical sequences; red and green letters indicate where the genomic sequence encodes a different amino acid. Although the ADAM18 protein sequence has a length of 715 amino acids, only 16 amino acids align as a single block to the lizard genome.



(c)

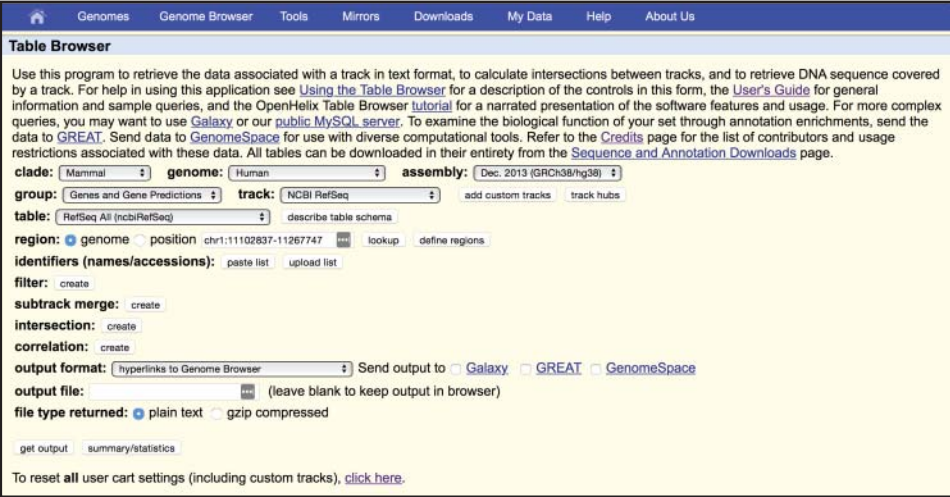
Figure 4.11 (Continued)

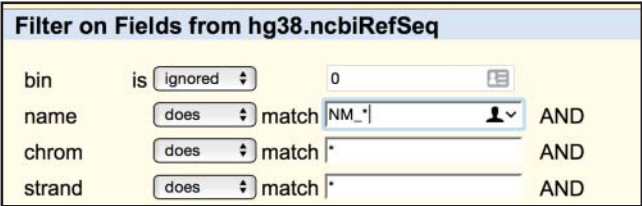
UCSC Table Browser

The Table Browser tool provides users a text-based interface with which to query, intersect, filter, and download the data that are displayed graphically in the Genome Browser. These data can then be saved in a spreadsheet for further analysis, or used as input into a different program. Using a web-based interface, users select a genome assembly, track, and position, then choose how to manipulate that track data and what fields to return. This example will demonstrate how to retrieve a list of all NCBI mRNA reference sequences that overlap with an SNP from the Genome-Wide Association Study (*GWAS*) *Catalog* track, which identifies genetic loci associated with common diseases or traits. The *GWAS* Catalog is a manually curated collection of published genome-wide association studies that assayed at least 100 000 SNPs, in which all SNP-trait associations have *p* values of $<1 \times 10^{-5}$ (Buniello et al. 2019).

The Table Browser landing page is accessible from either the UCSC Genome Browser home page or the *Tools* pull-down menu. First, reset all user cart settings by clicking on the *click here* link at the bottom of the Table Browser settings section.

Then, select the *NCBI RefSeq* track on the GRCh38 genome assembly (Figure 4.12a). Create a *filter* to limit the search to curated mRNA reference sequences in the NM_ accession series (Box 1.2; Figure 4.12b). Next, intersect the *RefSeq* track with variants from the *GWAS* Catalog (Figure 4.12c). Finally, on the Table Browser form, change the output format to *hyperlinks to Genome Browser*, then click *get output*. The output is a list of 3000+ RefSeq mRNAs that overlap with a variant from the *GWAS* Catalog (Figure 4.12d). The Genome Browser view of one of the transcripts, from the gene arginine–glutamic acid dipeptide (RE) repeats (RERE), and the six SNPs from the *GWAS* Catalog that it overlaps, can be found by clicking on the first link in the results list and is shown in Figure 4.12e.

(a) 

(b) 

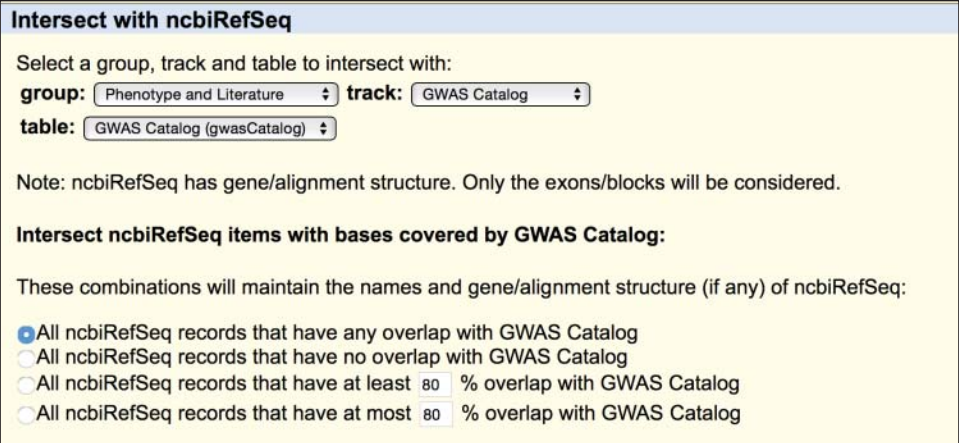
(c) 

Figure 4.12 Configuring the UCSC Table Browser. The link to the Table Browser is in the *Tools* menu at the top of each page. (a) On the Table Browser home page, first reset all previous selections by clicking on the reset button at the bottom of the window. Next, select the track called *NCBI RefSeq* in the group *Genes and Gene Predictions* on the human GRCh38 genome assembly. The *region* should be set to *genome* and the *output format* to *hyperlinks to Genome Browser*. (b) Create a *filter* to limit the search to curated mRNA reference sequences in the *NM_* accession series (see Box 1.2). Click on the *filter* button shown in Figure 4.12a and enter the term *NM_** in the *name* field. The asterisk is a wildcard character that matches any text. Thus, this setting will limit the results to those curated RefSeqs whose name contains the term *NM_*. (c) Create an intersection between the *RefSeq* track and the variants from the *GWAS Catalog*. Click on the *intersection* button shown in Figure 4.12a and select the appropriate track. The *group* is *Phenotype and Literature* and the *track* is called *GWAS Catalog*. Leave other selections set to the default. (d) Click on the *get output* button shown in Figure 4.12a. The output is a list of more than 3000 RefSeq mRNAs that overlap with a variant from the *GWAS Catalog*. Each RefSeq is hyperlinked to the *Genome Browser*. (e) The first link is to *NM_001042682.1*, a transcript of the gene arginine–glutamic acid dipeptide (*RE*) repeats (*RERE*). The genomic context of *RERE* shows the eight SNPs from the *GWAS Catalog* that it overlaps.

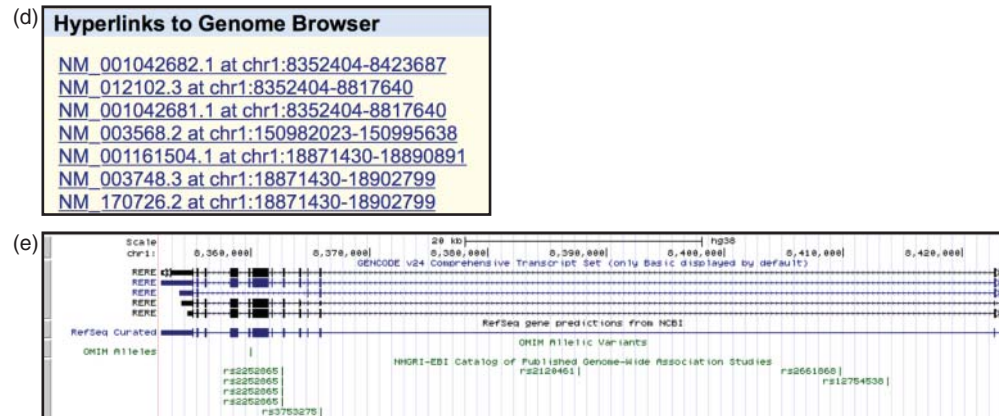


Figure 4.12 (Continued)

UCSC also provides a related tool called the *Data Integrator*. The *Data Integrator* has a more sophisticated intersection function than does the Table Browser, as it can intersect data from up to five separate tracks, and output fields from both the selected tracks and related tables. Thus, for example, output from the *Data Integrator* could include the gene symbol in addition to the accession number for each transcript on the *RefSeq* track, along with the dbSNP identifier for the variants in the GWAS Catalog. However, the *Data Integrator* does not allow for filtering, so it is not possible to restrict the output to only RefSeq mRNA genes.

ENSEMBL Genome Browser

The Ensembl Genome Browser (Cunningham et al. 2019) got its start in 1999 (Hubbard et al. 2002) with the display of the human genome assembly. Like the UCSC Genome Browser, it has grown significantly over the years. The main Ensembl site focuses on vertebrates and includes assemblies from almost 90 species. Ensembl has also created specialized sibling databases for other groups of organisms, including EnsemblPlants (nearly 50 species), EnsemblMetazoa (nearly 70 species), EnsemblProtist (more than 100 species), and EnsemblFungi (more than 800 species), and the very large EnsemblBacteria, with around 44 000 species. The amount of available genome data and annotations varies by organism, but the general browser navigation principles are the same for all. An additional resource is Pre!Ensembl, which displays genomes that are in the process of being annotated. Genomes on this site have an assembly and BLAST interface but, for the most part, no gene predictions.

Like the UCSC Genome Browser, the Ensembl Browser makes available multiple versions of genome assemblies. Integrated into the assemblies may be gene, genome variation, gene regulation, and comparative genomics annotation. Annotations are organized as sets of tracks. Ensembl incorporates data from a variety of public sources, including NCBI, UCSC, model organism databases, and more, and updates data and software in a formal release process, which can be tracked by release number. Importantly, previous Ensembl releases are archived on the web site and are available for view. Thus, even after a genome assembly or annotation set has been updated, it is possible to view the older data using all the regular functions of the Ensembl web site. This archive process sets Ensembl apart from UCSC, where the genome assembly remains stable, but the annotations may change on a weekly basis. Each Ensembl page has a link at the bottom called *View in archive site*. The archive site provides links to older versions of that page, including previous annotation sets on the same genome assembly, as well as prior genome assemblies.

The Ensembl Browser provides many of the same types of resources and tools as does the UCSC Genome Browser. Sequences can be aligned to the assembled genomes using either

BLAT or BLAST, and data can be returned in various tabular formats using BioMart (Kinsella et al. 2011). Data and software can be retrieved from the *Downloads* menu, available from most browser pages. In the *Tools* menu, Ensembl provides a number of additional tools to manipulate data, including the Variant Effect Predictor (VEP) (McLaren et al. 2016), which predicts functional consequences of known and unknown variants, File Chameleon, which reformats files available on the Ensembl FTP site, and Assembly Converter, which is like UCSC's liftOver and is used to convert coordinates between genome assemblies. The *Help & Documentation* menu provides substantial written and video-based information about how to navigate and interpret the Ensembl site, far beyond the level of detail presented in this chapter.

Ensembl also provides ways for users to upload their data into the browser. Properly formatted tracks can be added to the display by selecting the *Custom tracks* option from the left side of any species-specific page. The data can be uploaded to Ensembl from a file on the user's computer or, if it is saved on a web server, the browser can read it from a URL. Users who create an account at Ensembl can save track data to the Ensembl database server and view them later from any computer. To share custom tracks or even a customized view of the Genome Browser

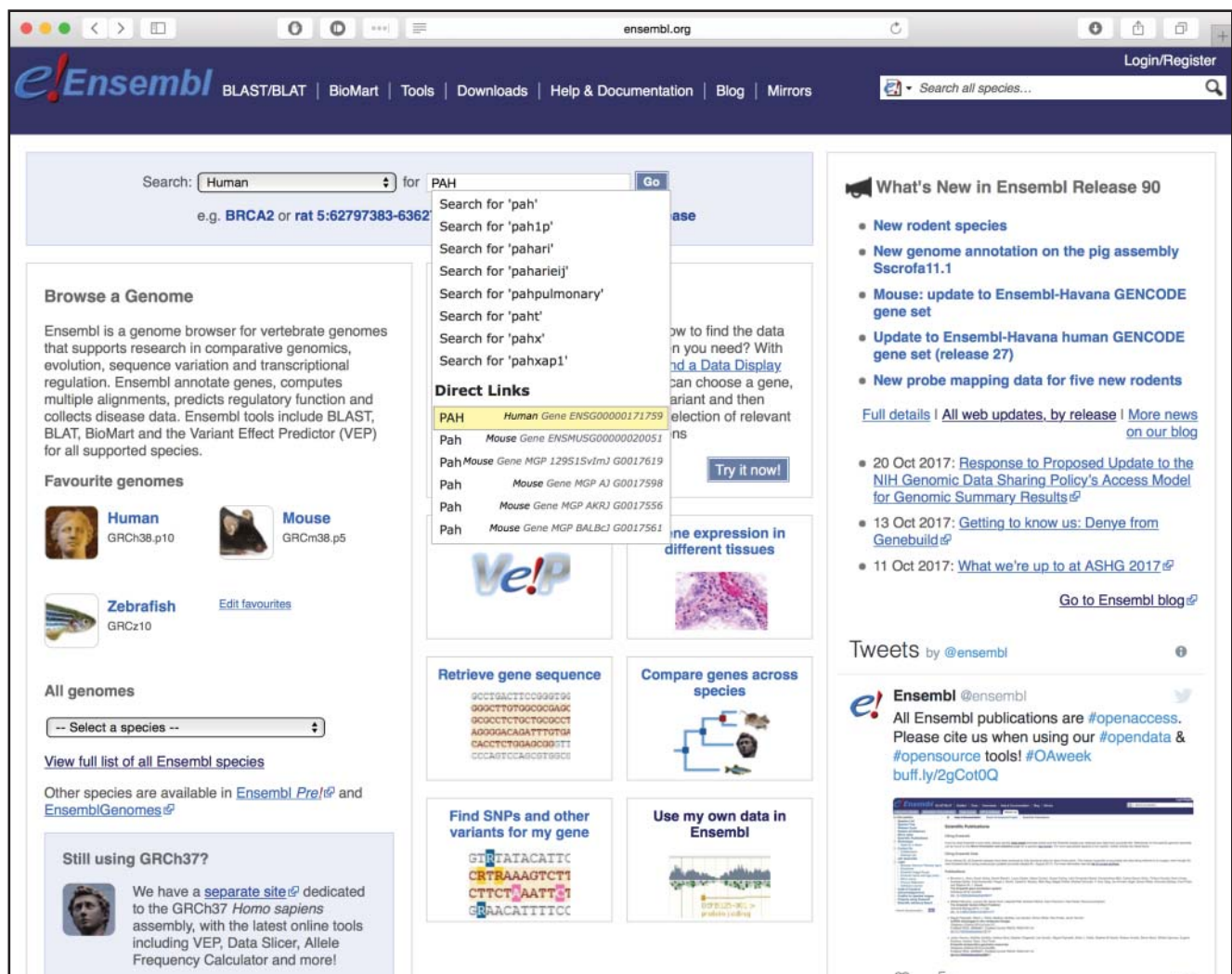


Figure 4.13 The home page of the Ensembl Genome Browser, showing a query for the human gene *PAH*. The browser suggests results based on the search term submitted. By default, the search box interfaces with the most recent version of the genome assembly, GRCh38, at the time of this writing. A link to the previous human genome assembly, GRCh37, is provided at the bottom of the page. Older assemblies from other organisms are available in the Ensembl archives.

The screenshot shows the Ensembl genome browser interface for the human *PAH* gene. The top navigation bar includes links for BLAST/BLAT, BioMart, Tools, Downloads, Help & Documentation, Blog, and Mirrors. A search bar is located in the top right corner. The main content area is divided into several sections:

- Gene-based displays:** A sidebar menu on the left lists various views such as Summary, Splice variants, Transcript comparison, Gene alleles, Sequence, Secondary Structure, Comparative Genomics, Genomic alignments, Gene tree, Gene gain/loss tree, Orthologues, Paralogues, Ensembl protein families, Ontologies, Genetic Variation, Variant table, Variant image, Structural variants, Gene expression, Regulation, External references, Supporting evidence, ID History, and Gene history.
- Gene: PAH ENSG00000171759:** The main content area displays the gene name and its description: phenylalanine hydroxylase [Source:HGNC Symbol;Acc:HGNC:8582].
- Description:** PH
- Synonyms:** PH
- Location:** Chromosome 12: 102,836,885-102,958,410 reverse strand. GRCh38:CM000674.2
- About this gene:** This gene has 18 transcripts (splice variants), 81 orthologues, 3 paralogues, is a member of 1 Ensembl protein family and is associated with 8 phenotypes.
- Transcripts:** A button labeled "Show transcript table" is visible.
- Summary:** A section with a question mark icon, containing details such as Name (PAH (HGNC Symbol)), CCDS (This gene is a member of the Human CCDS set: CCDS9092.1), UniProtKB (This gene has proteins that correspond to the following UniProtKB identifiers: P00439), Ensembl version (ENSG00000171759.9), Other assemblies (This gene maps to 103,230,663-103,352,188 in GRCh37 coordinates. View this locus in the GRCh37 archive: ENSG00000171759), Gene type (Protein coding), and Annotation method (Annotation for this gene includes both automatic annotation from Ensembl and Havana manual curation, see article).

At the bottom of the page, there is a button labeled "Go to Region in Detail for more tracks and navigation options (e.g. zooming)".

Figure 4.14 The *Gene* tab for the human *PAH* gene. This landing page provides links to many gene-specific resources.

with colleagues, click on the *Share this Page* link on the left sidebar. Ensembl also supports *Track Hubs*, both public ones that are registered on the EMBL-EBI Track Hub Registry as well as private ones.

Like the UCSC Genome Browser home page, the home page of Ensembl is a stepping-off point for many Ensembl resources. Links to commonly used tools, such as BLAST and BLAT, are provided on the top and middle sections of the page, and recent data updates are highlighted in the right column. The home page for each genome can be accessed by selecting the organism name in the pull-down menu in the *Browse a Genome* section in the center of the page. A search box at the top of the page provides access to Ensembl. To search for the human *PAH* gene, select Human from the pull-down menu and type the term *PAH* in the search box. Ensembl will provide several suggested hits, including a direct link to the human *PAH* gene (Figure 4.13).

Ensembl data displays are organized in tabs. The *Gene* tab (Figure 4.14) has links to a number of gene-specific views and resources. For example, from the index on the left side of the *Gene* tab view, the *Comparative Genomics* → *Orthologues* link lists the computationally predicted orthologs of the selected gene that Ensembl has identified among the available genome assemblies (Herrero et al. 2016; Figure 4.15). The *Location* tab provides a graphical view of the genomic context of the gene, similar to the view available at UCSC. The link to the *Location* tab is at the top of the *Gene* tab view in Figure 4.14. The *Location* tab view is shown in Figure 4.16 and depicts, at three different zoom levels, the genomic context of the *PAH* gene on the GRCh38 genome assembly. The *PAH* gene has been mapped to chromosome 12, and the top panel shows a cartoon of that chromosome, with the region surrounding the *PAH* gene outlined in a red box. This red box is expanded in the middle panel of the figure, which shows ~1 Mb of chromosome 12 around the *PAH* gene. The genes are shown as colored blocks, with their identifiers noted below them. The region outlined in red in this middle section is further

Gene: PAH ENSG00000171759

Description phenylalanine hydroxylase [Source:HGNC Symbol;Acc:HGNC:8582]#

Synonyms PH

Location [Chromosome 12: 102,836,885-102,958,410 reverse strand.](#)
GRCh38:CM000674.2

About this gene This gene has 18 transcripts ([splice variants](#)), 81 orthologues, 3 paralogues, is a member of 1 [Ensembl protein family](#) and is associated with 8 phenotypes.

Transcripts [Show transcript table](#)

Orthologues [Download orthologues](#)

Summary of orthologues of this gene [Hide](#)

Click on 'Show details' to display the orthologues for one or more groups of species. Alternatively, click on 'Configure this page' to choose a custom list of species.

Species set	Show details	With 1:1 orthologues	With 1:many orthologues	With many:many orthologues	Without orthologues
Primates (11 species) Humans and other primates	<input type="checkbox"/>	11	0	0	0
Rodents and related species (23 species) Rodents, lagomorphs and tree shrews	<input type="checkbox"/>	23	0	0	0
Laurasiatheria (14 species) Carnivores, ungulates and insectivores	<input type="checkbox"/>	14	0	0	0
Placental Mammals (53 species) All placental mammals	<input type="checkbox"/>	53	0	0	0
Sauropsida (7 species) Birds and Reptiles	<input type="checkbox"/>	7	0	0	0
Fish (11 species) Ray-finned fishes	<input type="checkbox"/>	10	0	0	1
All (83 species) All species, including invertebrates	<input type="checkbox"/>	80	0	0	3

Selected orthologues [Hide](#)

Species	Type	Orthologue	dN/dS	Target %id	Query %id	GOC Score	WGA Coverage	High Confidence
Alpaca (<i>Vicugna pacos</i>)	1-to-1	PAH (ENSYPAG00000002017) View Gene Tree Compare Regions (GeneScaffold_1527:799,923-873,069:-1) View Sequence Alignments	n/a	88.27 %	88.27 %	100	100.00	Yes
Amazon molly (<i>Poecilia formosa</i>)	1-to-1	pah (ENSPFOG00000018041) View Gene Tree Compare Regions (KI519679.1:2,325,684-2,342,763:-1)	n/a	71.18 %	72.12 %	50	4.91	Yes

Figure 4.15 Computationally predicted orthologs of the human *PAH* gene, from the *Comparative Genomics* → *Orthologues* link in Figure 4.14. Ensembl provides a detailed analysis of the orthologs calculated for each gene. Orthologs are grouped by species, such as primates, rodents, and sauropsids. Links to individual orthologs are shown at the bottom of the page.

expanded in the large bottom panel, which zooms in on the *PAH* gene itself. Individual tracks are visible in this view. Note the track called *Contigs*, a blue bar that represents the underlying assembled contigs. By convention, any transcripts shown above this track are transcribed from left to right. Transcripts drawn below the *Contigs* track, such as the *PAH* transcripts, are transcribed on the opposite strand, from right to left.

The default human gene set used by Ensembl is the GENCODE Comprehensive set (Box 4.2). Ensembl displays 18 *PAH* isoforms, each with a slightly different pattern of exons (Figure 4.16). Coding exons are depicted as solid blocks, non-coding exons as outlined blocks, and introns are the lines that connect them. The transcripts are color coded to indicate their status: gold transcripts are protein coding and have been annotated by both the Ensembl and HAVANA team at the WTSI, red transcripts are protein coding and have been annotated by either Ensembl or HAVANA, and blue transcripts are processed transcripts that are non-protein coding. Clicking on a transcript pops up a box with additional information about that feature, including its accession number, and, for a transcript, the transcript type and gene prediction source (Box 4.4; Figure 4.16).

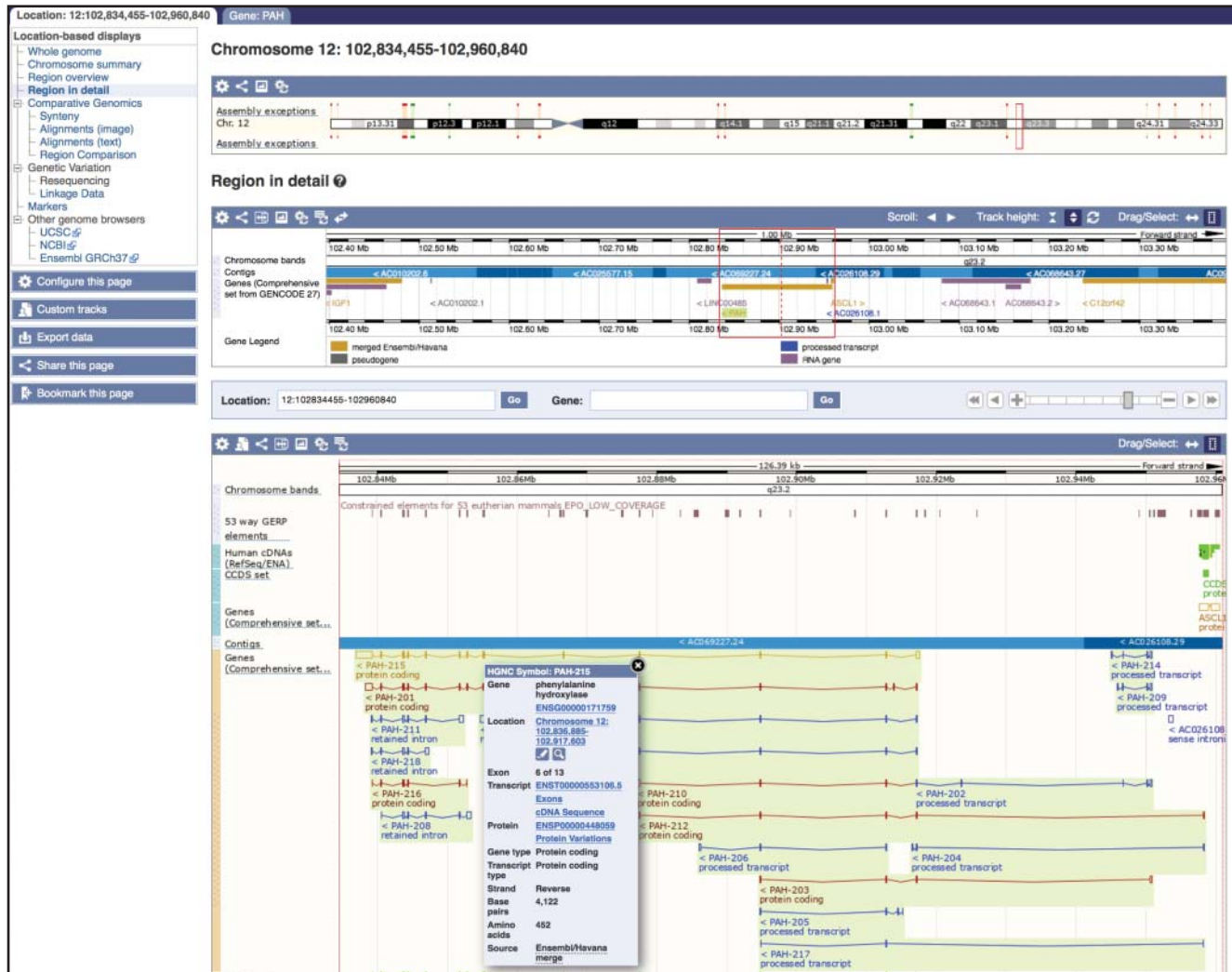


Figure 4.16 The *Location* tab for the human *PAH* gene. The *Location* tab is divided into three sections. The top section shows a cartoon of human chromosome 12, with the region surrounding the *PAH* gene outlined in a red box. Other red and green lines on the cartoon indicate assembly exceptions, or regions of alternative sequence that differ from the primary assembly because of allelic sequence or incorrect sequence, as determined by the Genome Reference Consortium. The *Region in detail* shows a zoomed-in view of the region outlined by the red box in the top section of the page. Genes are indicated by rectangles, colored as described in the gene legend below the graphic. The gene identifiers, along with the direction of transcription, are shown below the rectangles. The bottom section shows a zoomed-in view of the region surrounded by the red box in the *Region in detail*. The blue bar represents the genomic contig in this region. In the *Genes* track, genes above the bar are transcribed from left to right; those below the contig are transcribed from right to left. A few of the *PAH* transcripts, which are transcribed from right to left, are visible in this view. Gold transcripts are merged HAVANA/Ensembl transcripts; red are Ensembl protein-coding transcripts; blue transcripts are non-protein-coding processed transcripts. The pop-up display, activated when clicking on a particular transcript, shows the details for the first transcript in the *Genes* track, *PAH-215*.

Box 4.4 Ensembl Stable IDs

Ensembl assigns accession numbers to many data types in its database. Each identifier begins with the organism prefix; for human, the prefix is ENS; for mouse, it is ENSMUS; and for anole lizard, it is ENSACA. Next comes an abbreviation for the feature type: G for gene, T for transcript, P for protein, R for regulatory, and so forth. This is followed by a series of digits, and an optional version. The version number increments when there is a change

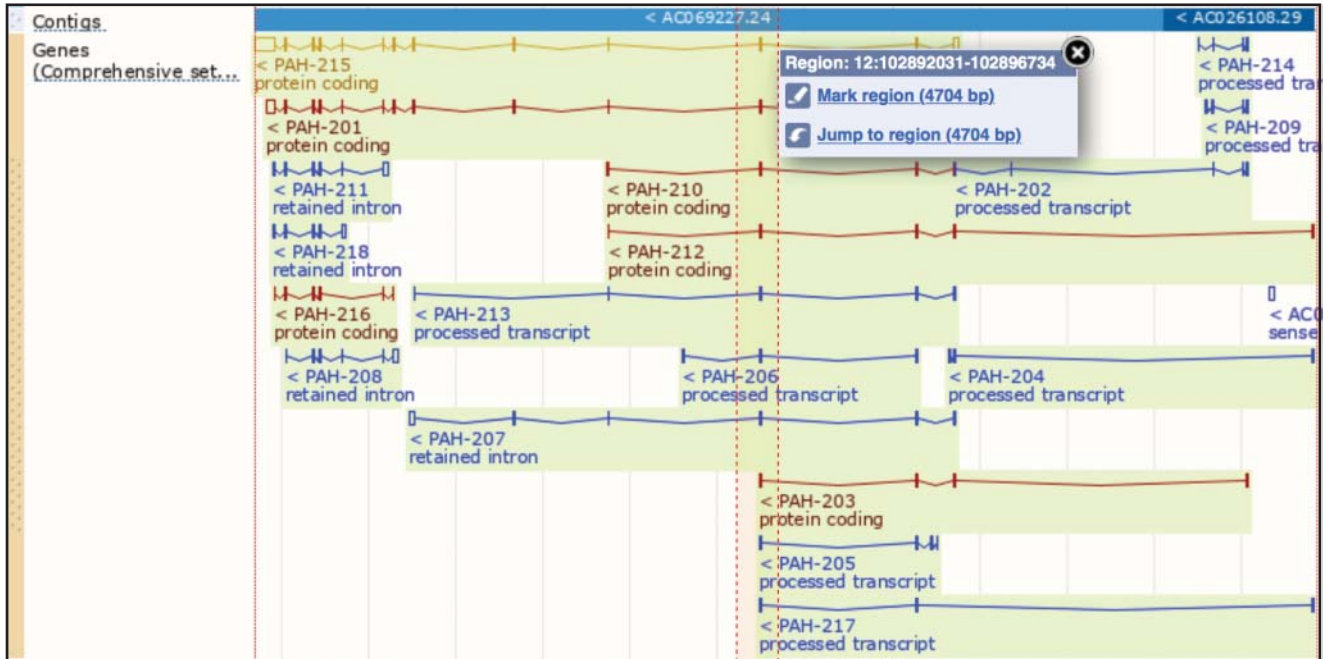
in the underlying data. The gene version changes when the underlying transcripts are updated, and the transcript and protein versions increment when the sequence changes.

For example, the human *PAH* gene has the following identifiers:

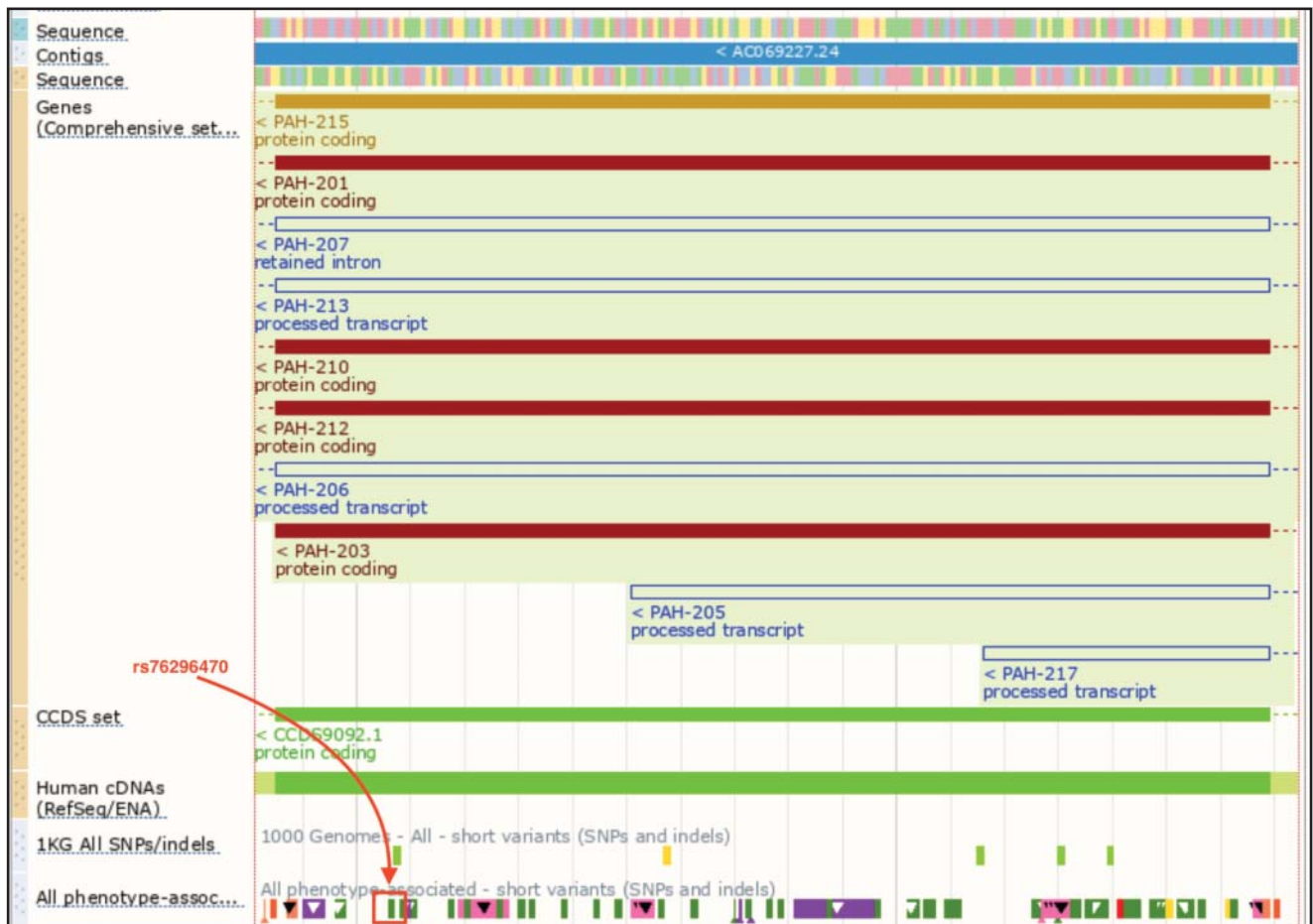
ENSG00000171759 . 9: the identifier of the human *PAH* gene
 ENST00000553106 . 5: the identifier of one transcript of the human *PAH* gene,
 transcript PAH-215
 ENSP00000448059 . 1: the identifier of the protein translation of
 transcript PAH-215, ENST00000553106 . 5
 ENSR00000056420: the identifier of a promoter of several PAH transcripts

Navigation controls between the second and third panels of the *Location* tab allow the display to be zoomed or moved to the left or right. The blue bar at the top of the *Region in detail* allows users to toggle between *Drag* and *Select*. When the *Drag* option is highlighted, click on the graphical view window and drag it to the left or right to change the location. When the *Select* option is highlighted, click on a region of interest in the graphical view, then, holding the mouse button down, scroll to the left or right to highlight the region (Figure 4.17a). The highlight can be left on for visualization purposes or, alternatively, select *Jump to region* to zoom in to the selected region. Figure 4.17b shows the results of zooming in to the last exon of transcript *PAH-203*; since the gene is transcribed from right to left, the last exon is on the left. Note the track called *All phenotype-associated short variants (SNPs and indels)* that contains those variants that have been associated with a phenotype or disease. SNPs are color coded by function, with dark green indicating coding sequence variants. Select the dark green SNP, highlighted with a red box near the left end of the window, and follow the link for additional information. The resulting *Variants* tab provides links to SNP-related resources. For example, the *Phenotype Data* for this SNP (rs76296470; Figure 4.18a) shows that this variant is pathogenic and is associated with the disease phenylketonuria. The most severe consequence for this SNP is a *stop gained*. Further details about the consequences are available under the *Genes and regulation* link (Figure 4.18b) on the left sidebar. This variant is found in 10 transcripts of the *PAH* gene. In five of those transcripts, it alters one nucleotide in a codon, changing an arginine to a stop codon, thus truncating the PAH protein. In the other five transcripts, either the variant is downstream of the gene or the transcript is non-coding.

Ensembl makes available many annotation tracks through the *Configure this page* link on the left sidebar. There are over 500 tracks available for display on GRCh38, with the majority falling in the categories of Variation, Regulation, and Comparative Genomics. The *Ensembl Regulatory Build* includes regions that are likely to be involved in gene regulation, including promoters, promoter flanking regions, enhancers, CCCTC-binding factor (CTCF) binding sites, transcription factor binding sites (TFBS), and open chromatin regions (Zerbino et al. 2016). A summary *Regulatory Build* track is turned on by default in the *Location* tab, and the display of individual features can be adjusted in the *Configure this page* menu. In the UCSC Genome Browser, the *GTEX* track shows that the *PAH* gene is highly expressed in liver and kidney (Figure 4.10); the epigenetic factors that may be controlling this activity can be viewed in *Ensembl Regulatory Build*. To view these factors, navigate to *Regulation* → *Histones & polymerases* on the *Configure this page* menu, mouse over the *HepG2* human liver carcinoma line, and select *All features* for *HepG2* (Figure 4.19a). In addition, navigate to *Regulation* → *Open chromatin & TFBS* and confirm that the *DNase1* track is in its default state for *HepG2*; the dark blue indicates that the track is *shown*. Close the *Configure this page* menu by clicking on the check mark in the upper right corner of the pop-up window. Notice that the *Regulatory Build* track has now expanded to include the selected gene regulatory marks in the *HepG2* cell line. Zoom in on the first exon of transcript *PAH-215* to see the promoter region of this gene, being mindful of the orientation of the gene (Figure 4.19b). The solid red rectangle in the *Regulatory*

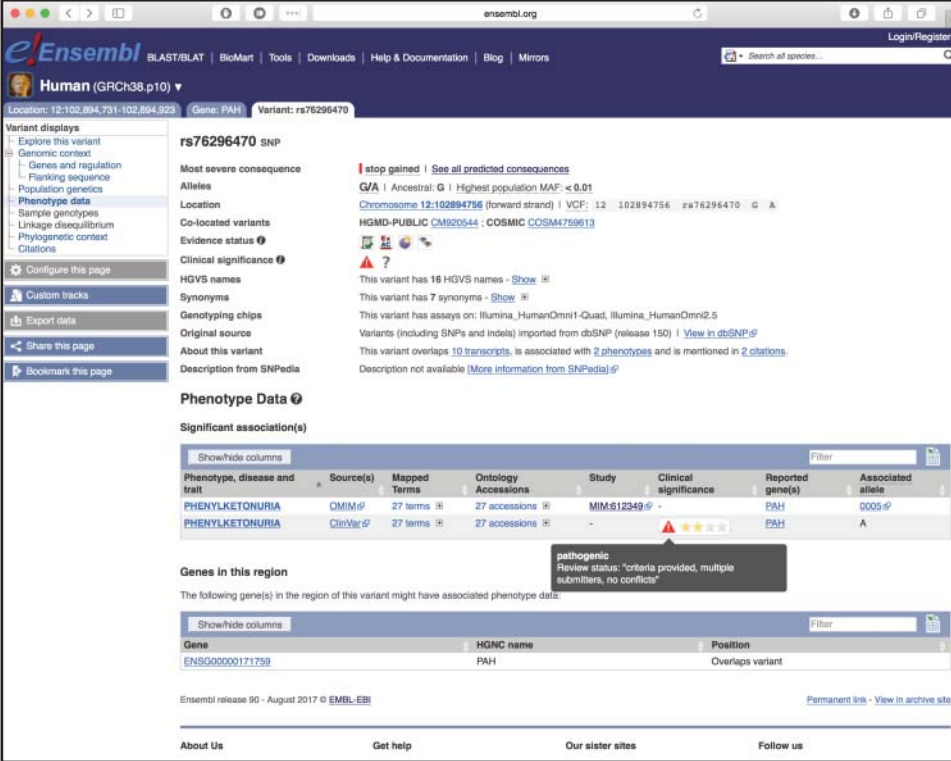


(a)



(b)

Figure 4.17 Zooming in on the bottom section of the *Location* tab from Figure 4.16. (a) Highlight a region of interest, the final exon of *PAH* transcript *PAH-203*, by clicking the mouse and then scrolling to the left or right. In order to highlight the region, the *Drag/Select* toggle in the blue bar at the top of the section must first be set to *Select*. (b) To zoom in to the highlighted region, select *Jump to region*. It may take a few iterations to create the view in this figure. At the bottom of the window is a track labeled *All phenotype-associated - short variants (SNPs and indels)*. In this track, the SNP rs76296470 has been manually highlighted in red.

(a) 

rs76296470 SNP

Most severe consequence: **stop gained** | See all predicted consequences

Alleles: **G/A** | Ancestral: G | Highest population MAF: < 0.01

Location: **Chromosome 12:102894756** (forward strand) | VCF: 12 102894756 rs76296470 G A

Co-located variants: **HGMD-PUBLIC CM920544** ; **COSMIC COSM4759613**

Evidence status: **7**

Clinical significance: **pathogenic**
Review status: "criteria provided, multiple submitters, no conflicts"

Phenotype Data

Phenotype, disease and trait	Source(s)	Mapped Terms	Ontology Accessions	Study	Clinical significance	Reported gene(s)	Associated allele
PHENYLKETONURIA	OMIM	27 terms	27 accessions	MIM812349		PAH	0005
PHENYLKETONURIA	ClinVar	27 terms	27 accessions			PAH	A

Genes in this region

Gene	HGNC name	Position
ENSG00000171759	PAH	Overlaps variant

(b) **Genes and regulation**

Gene and Transcript consequences

Gene	Transcript (strand)	Allele (transcript allele)	Consequence Type	Position in transcript	Position in CDS	Position in protein	Amino acid	Codons	SIFT	PolyPhen	Detail
ENSG00000171759 HGNC: PAH biotype: protein_coding	ENST00000553106.5 (-)	A (T)	stop gained	804 (out of 4122)	331 (out of 1359)	111 (out of 452)	R ^H	CGA/TGA	-	-	Show
ENSG00000171759 HGNC: PAH biotype: processed_transcript	ENST00000635500.1 (-)	A (T)	downstream gene variant	-	-	-	-	-	-	-	Show
ENSG00000171759 HGNC: PAH biotype: processed_transcript	ENST00000548928.1 (-)	A (T)	non coding transcript exon variant	253 (out of 472)	-	-	-	-	-	-	Show
ENSG00000171759 HGNC: PAH biotype: processed_transcript	ENST00000551988.5 (-)	A (T)	non coding transcript exon variant	420 (out of 584)	-	-	-	-	-	-	Show
ENSG00000171759 HGNC: PAH biotype: protein_coding	ENST00000307000.7 (-)	A (T)	stop gained	587 (out of 2466)	316 (out of 1344)	106 (out of 447)	R ^H	CGA/TGA	-	-	Show
ENSG00000171759 HGNC: PAH biotype: protein_coding	ENST00000546844.1 (-)	A (T)	stop gained	684 (out of 705)	331 (out of 352)	111 (out of 117)	R ^H	CGA/TGA	-	-	Show
ENSG00000171759 HGNC: PAH biotype: protein_coding	ENST00000550979.6 (-)	A (T)	stop gained	315 (out of 652)	316 (out of 449)	106 (out of 149)	R ^H	CGA/TGA	-	-	Show
ENSG00000171759 HGNC: PAH biotype: protein_coding	ENST00000551337.5 (-)	A (T)	stop gained	598 (out of 675)	331 (out of 408)	111 (out of 136)	R ^H	CGA/TGA	-	-	Show
ENSG00000171759 HGNC: PAH biotype: retained_intron	ENST00000549111.5 (-)	A (T)	non coding transcript exon variant	427 (out of 1252)	-	-	-	-	-	-	Show
ENSG00000171759 HGNC: PAH biotype: processed_transcript	ENST00000548677.2 (-)	A (T)	downstream gene variant	-	-	-	-	-	-	-	Show

Figure 4.18 The Ensembl Variant tab. (a) To get more details about SNP rs76296470, click on the dark green SNP that is highlighted in red in the *All phenotype-associated – short variants (SNPs and indels)* track in Figure 4.17b. On the pop-up menu, click on *more about rs76296470*. The *Phenotype Data* section of the *Variant* tab is available from the link in the blue sidebar. This variant is pathogenic for phenylketonuria. (b) The *Genes and regulation* section of the *Variant* tab shows the location and function of the variant in the transcripts that overlap it. Depending on the transcript, the SNP can change a codon to a stop codon (stop gained), map downstream of a gene, or map to a non-coding transcript. The transcripts in this view represent alternatively spliced forms of the gene *PAH*.

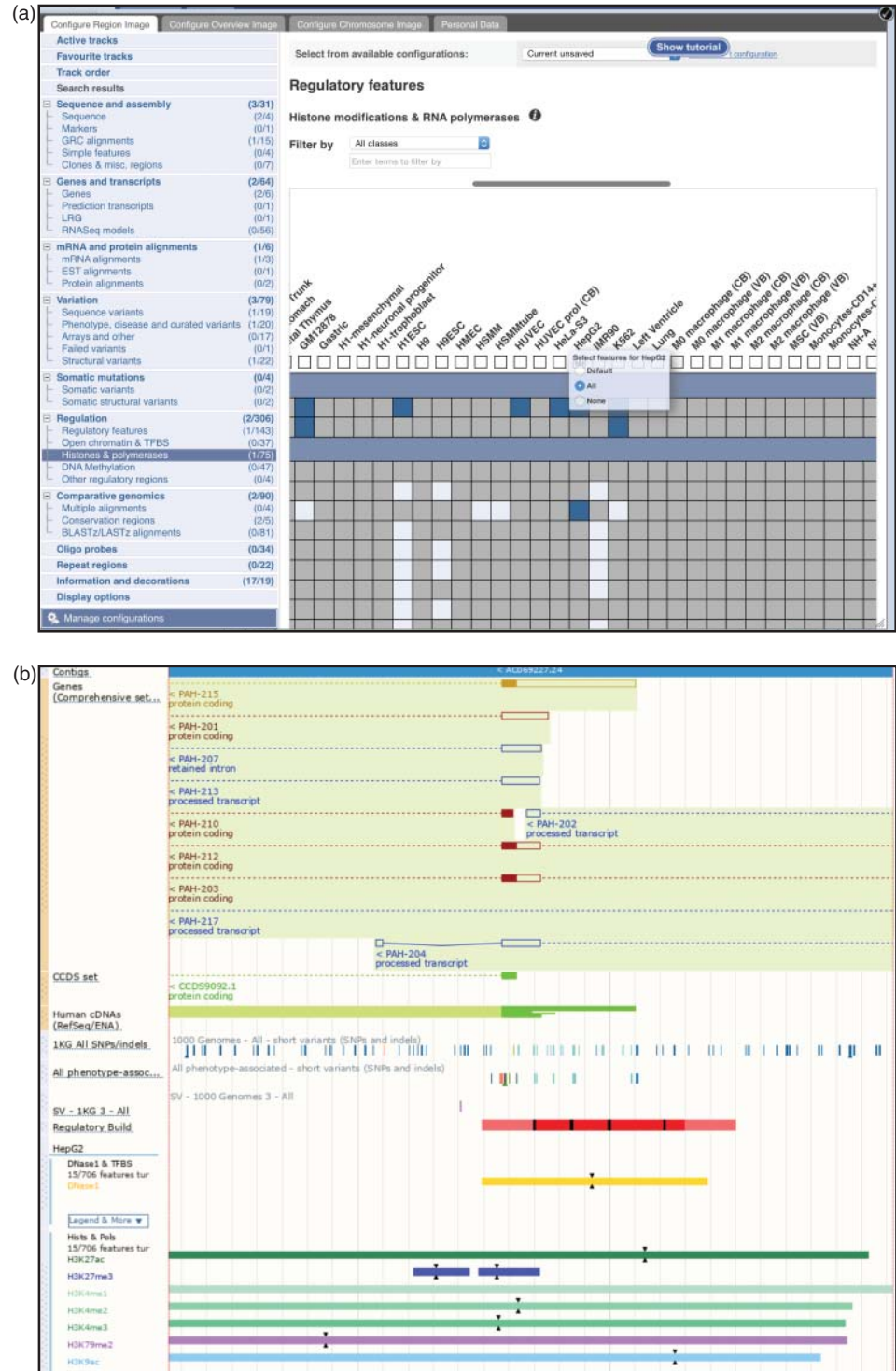
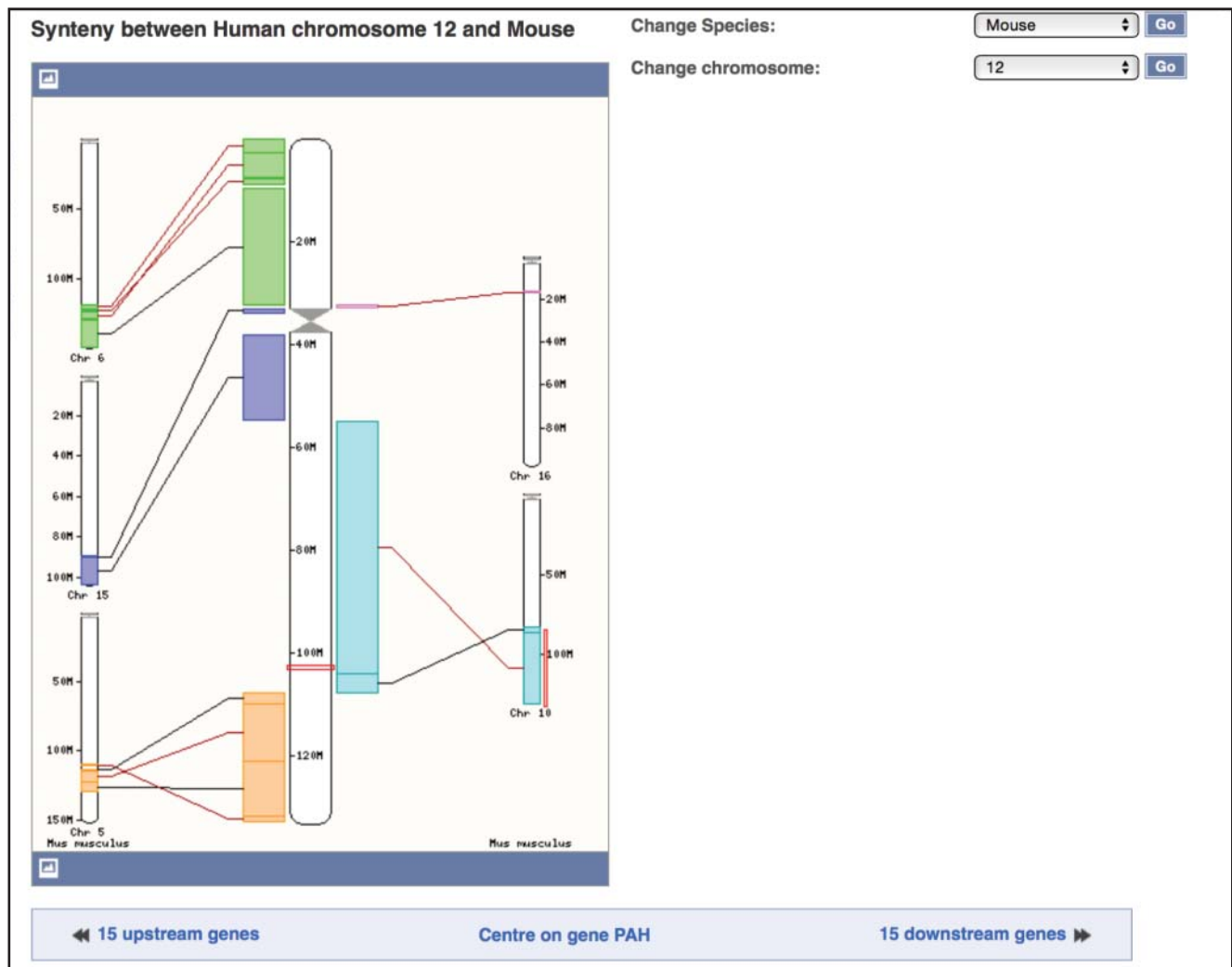


Figure 4.19 The Ensembl *Regulatory Build* track. (a) Go to *Configure this page* on the left side of the *Location* tab and select *Regulation* → *Histones & polymerases*. Scroll to the right to find the HepG2 (human liver cancer) cell type. Mouse over the text *HepG2* and turn on all features. Clicking on the box under the cell type will change the track style; leave that set to the default of *Peaks*. Click on the black check mark on the upper right corner of the configuration window to save the settings and exit the setup. To turn on the *DNase1* (DNaseI hypersensitive sites track), select *Regulation* → *Open chromatin & TFBS* and ensure that the *DNase1* box in the HepG2 column is colored dark blue so that it is in the *Shown* configuration. Click on the black check mark on the upper right corner of the configuration window to save the settings again. (b) Back on the *Region in detail* section of the *Location* tab, zoom in to the first exon of transcript PAH-215. Note that the first exon is on the right end of the transcript, as the gene is transcribed from right to left. The resulting display shows the details of the *Regulatory Build* track. The figure legend (not shown) explains that the solid red box is a promoter. The DNaseI hypersensitive site and histone marks are also shown as colored boxes.

Build track shows the location of the *PAH* promoter. The presence of a DNaseI hypersensitive site along with the activating histone marks of H3K27Ac, H3K4me1, H3K4me2, H3K4me3, H3K79me2, and H3K9Ac may help to explain why this gene is highly expressed in liver cells (Box 4.3). Detailed information about features in the *Regulatory Build* track, such as the source of the data, is available under the *Regulation* tab. Click on the feature and select its identifier (the letters ENSR, followed by numbers) to open this tab.

The left sidebar of the *Location* tab links to a number of additional useful resources. One of those, *Comparative Genomics* → *Synten* displays blocks of synteny between the human chromosome featured in the *Location* tab and chromosomes from about 30 different organisms. In these syntenic blocks, the order of genes and other sequence features is conserved



(a)

Figure 4.20 The Synteny view at Ensembl. (a) An overview of the syntenic blocks shared between human chromosome 12 and the mouse genome. The human chromosome is drawn in the middle of the display as a thick white box. The syntenic mouse chromosomes are represented by thinner white boxes along the side. The colored rectangles highlight regions of synteny between the human and mouse. A red outline illustrates the position of the *PAH* gene on the blue region of human chromosome 12 and on the blue region of mouse chromosome 10. (b) The *Location* tab for the *PAH* gene showing both the human and mouse syntenic regions. This is similar to the three-panel location tab shown in Figure 4.16, except that both the human and mouse genomes are depicted. The top panel (not shown) displays the full length human chromosome 12 and mouse chromosome 10. The second panel shows an overview of the genes in the region. The third panel focuses in on the *PAH* gene. Note that the regions in human and mouse appear to be presented in opposite orientations; in human, the *PAH* and *IGF1* genes are both transcribed from right to left, while in mouse they are transcribed from left to right.



(b)

Figure 4.20 (Continued)

across the genomes being compared. Figure 4.20a shows the synteny between human chromosome 12 and the mouse genome. A cartoon of the human chromosome 12 is shown in the center of the display as a thick white rectangle, and mouse chromosomes are drawn on the sides as thinner white rectangles. Colored rectangles indicate regions of synteny between the human and mouse. For example, the light blue region on human chromosome 12 is syntenic to the light blue region on mouse chromosome 10. The region surrounding the *PAH* gene is outlined in red on both human chromosome 12 and mouse chromosome 10. Below the cartoon is a list of the human genes and corresponding mouse orthologs in the region of *PAH*. Selecting *Region Comparison* next to one of the genes opens a new *Location* tab that depicts the syntenic human and mouse chromosomes stacked on top of each other so that surrounding features can be compared directly (Figure 4.20b). The upper panel shows the genomic context of the

PAH gene on human chromosome 12 (top) and mouse chromosome 10 (bottom). Note that the genes are transcribed in opposite directions, so the order of the surrounding genes is flipped. The bottom panel is zoomed in on the *PAH* gene itself. The *Regulatory Build* track on the mouse assembly shows several regulatory features in this region. Further inspection of the regulatory feature that overlaps with the 5' end of the mouse *Pah* gene reveals activating histone marks in liver and kidney cells, but not in other cell types (not shown), implying that the mouse *Pah* gene has similar expression patterns to its human ortholog. To reset the settings back to the default view, go to *Configure this page* in the left sidebar and select *Reset configuration*.

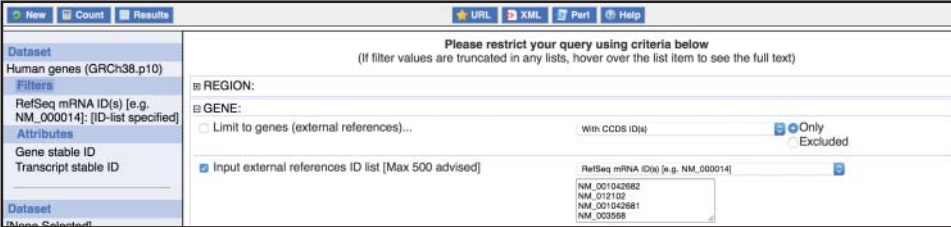
The Ensembl sequence data can also be queried via a BLAT or BLAST search by following the link at the top of any page. Earlier in this chapter, Figure 4.11 outlined how to use BLAT to look for a lizard homolog of the human *ADAM18* gene. Ensembl data can be searched by the more sensitive BLAST algorithm, including the TBLASTN program that is used to compare a protein query with a nucleotide database translated in all six reading frames. Copy and paste the FASTA-formatted protein sequence of NCBI RefSeq NP_001307242.1 into the *Sequence data* box on the BLAST page and carry out a TBLASTN search against the anole lizard genomic sequence. The sequence alignment of the top hit is shown in Figure 4.21. The human protein query is on the top line, and the translated lizard genomic sequence on the second. The sequences share only 32% sequence identity, but the alignment spans 650 amino acids, and some key sequence features are conserved; note the alignment of almost every cysteine residue. Thus, this lizard genomic sequence is indeed a homolog of human *ADAM18*. The BLAST algorithm, although about two orders of magnitude slower than BLAT for the same query, is able to find a lizard ortholog of the human protein.

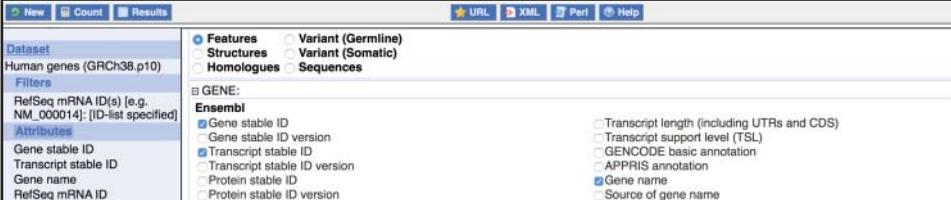
Ensembl Biomart

The BioMart tool at Ensembl is akin to the Table Browser at UCSC, in that it provides a web-based interface through which to access the data underlying the Ensembl Genome Browser. Results are returned as text or HTML-formatted tables. Ensembl hosts several mart databases that are described in the online documentation. The Ensembl Genes database contains the Ensembl gene set and integrates Ensembl genes, transcripts, and proteins with a number of resources, including external references, protein domains, sequences, variants, and homology data. After choosing a *Database* (e.g. Genes) and *Dataset* (genome assembly, e.g. *Homo sapiens*), the user specifies the *Filters* (basically, the input data) and the *Attributes* (the output data). Users can choose from among seven types of filters, including *Region* and *Gene*. A *Region* could be a chromosomal position, while a *Gene* could be an accession number, gene name, or even microarray probeset. The list of possible *Attributes* is long, and includes Ensembl data such as gene and transcript identifiers and positions, links to external data sources including RefSeq, UCSC, Pfam (protein families), and Gene Ontology (GO) terms, as well as mapping to orthologs in the Ensembl genome databases.

In this example, we will identify the mouse orthologs of the human mRNA reference sequences that are associated with common diseases or traits. To do this, we will start with the output of the UCSC Table Browser, the mRNA reference sequences that overlap with a variant from the GWAS Catalog, pull out the corresponding Ensembl gene and transcript identifiers, and then link to the mouse orthologs. The initial step is to retrieve the RefSeq accession numbers that overlap with a variant from the GWAS Catalog by reproducing the search shown in Figure 4.12d, this time changing the output format to *sequence*. Copy and paste the output from the Table Browser into your favorite text editor to create a list that contains only the accession numbers. Note that BioMart does not accept the accession.version format used by NCBI, so an accession number like NM_001042682.1 would need to be rewritten as NM_001042682.

At BioMart, the first step is to enter these accession numbers as *Filters* into the *Human Genes (GRCh38.p10) Dataset*. RefSeq mRNA accession numbers are entered in the filter called *Gene*

(a) 

(b) 

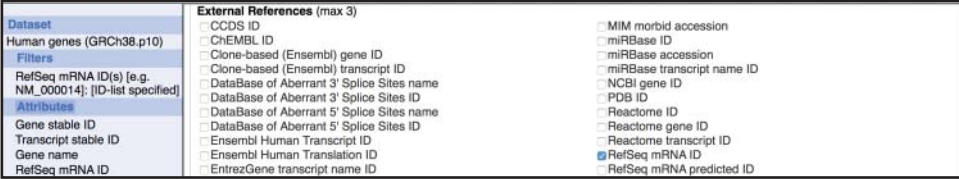
(c) 

Figure 4.22 Using BioMart to retrieve the mouse orthologs of the human RefSeqs from the GWAS Catalog. (a) Enter the input RefSeq accession numbers into BioMart. First, create a list of RefSeq accession numbers from the UCSC Table Browser output in Figure 4.12d. BioMart does not accept the accession.version format, so all of the text after the accession number itself will need to be removed. This step can be implemented using a text editor that can perform a wildcard search and replace. For example, to remove the period and all following text from each line, replace `.*` with an empty string. Although the resulting list of accession numbers will contain duplicates, as some RefSeqs have been mapped to alternate loci, any redundancy will be removed from the final BioMart results. At BioMart, click on *Filters* in the left sidebar, open the *Gene* menu, and click on *Input external references ID list*. In the pull-down menu, select *RefSeq mRNA IDs* as the type of identifier. Paste in the list of accession numbers, which should be of the form `NM_001042682`. Although BioMart instructions recommend limiting the number of access numbers to 500, the interface will process the 3000+ RefSeq accession numbers from the UCSC Table Browser output. (b) Set the BioMart *Attributes* (fields to be included in the output). Click on the *Attributes* in the left sidebar, select *Features* at the top of the page, then open the *Gene* menu. *Gene stable ID* and *Transcript stable ID* should be selected by default, and will return the Ensembl gene (ENSG) and transcript (ENST) identifiers. Also select *Gene name* to return the gene symbols (e.g. *ADAM18*). (c) Set additional *Attributes*. Close the *Gene* menu and open the *External* menu. Navigate to *External References* and select *RefSeq mRNA ID*. This step is needed to return the input RefSeq accession numbers so that they can be correlated later with the Ensembl identifiers. (d) BioMart output, including the identifiers requested above. Click on the *Results* button at the top of the page to retrieve the output. Check the box *Unique results only* to ensure that duplicated RefSeqs are returned only once. The order of the columns in the results file depends on the order in which the items were added to the list of *Attributes*. The net result is that each human RefSeq accession from the Table Browser is correlated with its Ensembl Gene and Transcript ID, as well as a gene symbol. (e) BioMart output, with human Ensembl Gene ID and gene symbol, as well as the orthologous mouse Ensembl Gene ID and gene symbol. Start a new query by clicking the *New* box at the top of the BioMart window. Select the same *Database*, *Dataset*, and *Filters* as before. Under *Attributes*, select the *Homologues* radio button. The human Ensembl Gene ID and gene symbol are in the *Gene → Ensembl* menu, called *Gene stable ID* and *Gene name*. The mouse Ensembl Gene ID and gene symbol are in the *Orthologues → Mouse Orthologues* menu, called *Mouse gene stable ID* and *Mouse gene name*. This step outputs the orthologous mouse Ensembl Gene ID and symbol for each human Ensembl Gene ID and symbol. The BioMart output from (d) and (e) can be merged to list the mouse ortholog of each human RefSeq from the GWAS Catalog (Figure 4.12d).

(c)

Dataset	Export all results to	File	tsv	<input type="checkbox"/> Unique results only	Go																																												
Human genes (GRCh38.p10)	Email notification to																																																
Filters	View	10	rows as	HTML	<input type="checkbox"/> Unique results only																																												
RefSeq mRNA ID(s) [e.g. NM_000014]; [ID-list specified]																																																	
Attributes	<table border="1" style="width: 100%; border-collapse: collapse; font-size: small;"> <thead> <tr> <th>Gene stable ID</th> <th>Transcript stable ID</th> <th>Gene name</th> <th>RefSeq mRNA ID</th> </tr> </thead> <tbody> <tr><td>ENSG00000156006</td><td>ENST00000288479</td><td>NAT2</td><td>NM_000015</td></tr> <tr><td>ENSG00000122971</td><td>ENST00000242592</td><td>ACADS</td><td>NM_000017</td></tr> <tr><td>ENSG00000129252</td><td>ENST00000305989</td><td>ADRB2</td><td>NM_000024</td></tr> <tr><td>ENSG00000162869</td><td>ENST00000294724</td><td>AGL</td><td>NM_000029</td></tr> <tr><td>ENSG00000172462</td><td>ENST00000307503</td><td>AGKT</td><td>NM_000030</td></tr> <tr><td>ENSG00000029534</td><td>ENST00000289734</td><td>ANK1</td><td>NM_000037</td></tr> <tr><td>ENSG00000110245</td><td>ENST00000227867</td><td>APOC3</td><td>NM_000040</td></tr> <tr><td>ENSG00000130203</td><td>ENST00000252498</td><td>APOE</td><td>NM_000041</td></tr> <tr><td>ENSG000000091583</td><td>ENST00000205848</td><td>APOH</td><td>NM_000042</td></tr> <tr><td>ENSG00000008103</td><td>ENST00000309740</td><td>FABP</td><td>NM_000043</td></tr> </tbody> </table>					Gene stable ID	Transcript stable ID	Gene name	RefSeq mRNA ID	ENSG00000156006	ENST00000288479	NAT2	NM_000015	ENSG00000122971	ENST00000242592	ACADS	NM_000017	ENSG00000129252	ENST00000305989	ADRB2	NM_000024	ENSG00000162869	ENST00000294724	AGL	NM_000029	ENSG00000172462	ENST00000307503	AGKT	NM_000030	ENSG00000029534	ENST00000289734	ANK1	NM_000037	ENSG00000110245	ENST00000227867	APOC3	NM_000040	ENSG00000130203	ENST00000252498	APOE	NM_000041	ENSG000000091583	ENST00000205848	APOH	NM_000042	ENSG00000008103	ENST00000309740	FABP	NM_000043
Gene stable ID	Transcript stable ID	Gene name	RefSeq mRNA ID																																														
ENSG00000156006	ENST00000288479	NAT2	NM_000015																																														
ENSG00000122971	ENST00000242592	ACADS	NM_000017																																														
ENSG00000129252	ENST00000305989	ADRB2	NM_000024																																														
ENSG00000162869	ENST00000294724	AGL	NM_000029																																														
ENSG00000172462	ENST00000307503	AGKT	NM_000030																																														
ENSG00000029534	ENST00000289734	ANK1	NM_000037																																														
ENSG00000110245	ENST00000227867	APOC3	NM_000040																																														
ENSG00000130203	ENST00000252498	APOE	NM_000041																																														
ENSG000000091583	ENST00000205848	APOH	NM_000042																																														
ENSG00000008103	ENST00000309740	FABP	NM_000043																																														
Dataset	[None Selected]																																																

(e)

Dataset	Export all results to	File	tsv	<input type="checkbox"/> Unique results only	Go																																												
Human genes (GRCh38.p10)	Email notification to																																																
Filters	View	10	rows as	HTML	<input type="checkbox"/> Unique results only																																												
RefSeq mRNA ID(s) [e.g. NM_000014]; [ID-list specified]																																																	
Attributes	<table border="1" style="width: 100%; border-collapse: collapse; font-size: small;"> <thead> <tr> <th>Gene stable ID</th> <th>Gene name</th> <th>Mouse gene stable ID</th> <th>Mouse gene name</th> </tr> </thead> <tbody> <tr><td>ENSG00000156006</td><td>NAT2</td><td>ENSMUSG00000051147</td><td>Nat2</td></tr> <tr><td>ENSG00000156006</td><td>NAT2</td><td>ENSMUSG00000056426</td><td>Nat3</td></tr> <tr><td>ENSG00000156006</td><td>NAT2</td><td>ENSMUSG000000025589</td><td>Nat1</td></tr> <tr><td>ENSG00000122971</td><td>ACADS</td><td>ENSMUSG00000029945</td><td>Acad9</td></tr> <tr><td>ENSG00000162869</td><td>ADRB2</td><td>ENSMUSG000000045791</td><td>Adrb2</td></tr> <tr><td>ENSG00000162869</td><td>AGL</td><td>ENSMUSG000000033400</td><td>Ag1</td></tr> <tr><td>ENSG00000172462</td><td>AGKT</td><td>ENSMUSG00000026272</td><td>Agkt</td></tr> <tr><td>ENSG00000029534</td><td>ANK1</td><td>ENSMUSG000000031543</td><td>Ank1</td></tr> <tr><td>ENSG00000110245</td><td>APOC3</td><td>ENSMUSG00000032081</td><td>Apoa3</td></tr> <tr><td>ENSG00000130203</td><td>APOE</td><td>ENSMUSG00000002982</td><td>Apoa</td></tr> </tbody> </table>					Gene stable ID	Gene name	Mouse gene stable ID	Mouse gene name	ENSG00000156006	NAT2	ENSMUSG00000051147	Nat2	ENSG00000156006	NAT2	ENSMUSG00000056426	Nat3	ENSG00000156006	NAT2	ENSMUSG000000025589	Nat1	ENSG00000122971	ACADS	ENSMUSG00000029945	Acad9	ENSG00000162869	ADRB2	ENSMUSG000000045791	Adrb2	ENSG00000162869	AGL	ENSMUSG000000033400	Ag1	ENSG00000172462	AGKT	ENSMUSG00000026272	Agkt	ENSG00000029534	ANK1	ENSMUSG000000031543	Ank1	ENSG00000110245	APOC3	ENSMUSG00000032081	Apoa3	ENSG00000130203	APOE	ENSMUSG00000002982	Apoa
Gene stable ID	Gene name	Mouse gene stable ID	Mouse gene name																																														
ENSG00000156006	NAT2	ENSMUSG00000051147	Nat2																																														
ENSG00000156006	NAT2	ENSMUSG00000056426	Nat3																																														
ENSG00000156006	NAT2	ENSMUSG000000025589	Nat1																																														
ENSG00000122971	ACADS	ENSMUSG00000029945	Acad9																																														
ENSG00000162869	ADRB2	ENSMUSG000000045791	Adrb2																																														
ENSG00000162869	AGL	ENSMUSG000000033400	Ag1																																														
ENSG00000172462	AGKT	ENSMUSG00000026272	Agkt																																														
ENSG00000029534	ANK1	ENSMUSG000000031543	Ank1																																														
ENSG00000110245	APOC3	ENSMUSG00000032081	Apoa3																																														
ENSG00000130203	APOE	ENSMUSG00000002982	Apoa																																														
Dataset	[None Selected]																																																

Figure 4.22 (Continued)

→ *Input external references ID list* (Figure 4.22a). The *Attributes* could be the Ensembl *Gene* and *Transcript* identifiers, as well as the *Gene name*, in the *Features* → *Gene* → *Ensembl* section (Figure 4.22b). To correlate the output with the RefSeq accession numbers entered as *Filters*, it is necessary to also select the RefSeq accession as an attribute, in the *Features* → *Gene* → *External References* section (Figure 4.22c). After the *Filters* and *Attributes* have been set, click on the *Results* button in the upper left to return the BioMart output (Figure 4.22d). Data can be returned as a text file or as a formatted page in the web browser, with hyperlinks to Ensembl resources. Because of the differences in gene annotation strategies, the mapping of NCBI RefSeq accession numbers to Ensembl gene and transcript identifiers is not one to one; some RefSeq accessions map to more than one Ensembl gene and/or transcript, and some Ensembl genes map to more than one RefSeq identifier.

Retrieving the mouse orthologs of the NCBI reference sequences must be done as a separate step, as it is not possible to return an external identifier (i.e. the starting RefSeq accession number) and an ortholog in the same BioMart query. Starting with the same *Filter* and human RefSeq accession numbers as before, choose the *Homologues* section of the *Attributes* and select the human Ensembl gene identifier and gene name under *Gene* → *Ensembl*, as well as the mouse Ensembl gene identifier and gene name under *Orthologues* → *Mouse Orthologues*. The results are shown in Figure 4.22e. Note that not all of the human gene identifiers have been mapped to a corresponding mouse ortholog. The goal of this exercise was to identify the mouse orthologs of the human RefSeq accession numbers from the GWAS Catalog. Using the human Ensembl gene identifiers as a key, the human RefSeq accession numbers can be added to the list of mouse orthologs. This can be carried out by using the VLOOKUP function in Microsoft Excel, or by writing a script in your favorite programming language, and is left as an exercise for the reader.

JBrowse

While the UCSC and Ensembl Genome Browsers provide user-friendly interfaces for viewing genomic data from well-characterized organisms, there are fewer applications for displaying genome assemblies and annotations for newly sequenced organisms or non-standard assemblies. The source code and executables for the UCSC Genome Browser are freely available for academic, non-profit, and personal use, and can be set up to display custom data, not just

those provided by UCSC. Thus, one option is for researchers to host their own UCSC Genome Browser and use it to share custom genomes with the bioinformatics community. An alternate method for sharing novel genome assemblies is to set up an *Assembly Hub*. Researchers host the specially formatted genomic sequence and data tracks on their own web site, and anyone with the URL can view the assembly through the UCSC Genome Browser.

Another way to share novel genome assemblies is to use JBrowse (Buels et al. 2016), a web-based genome browser that is part of the Generic Model Organism Database (GMOD) project, a suite of tools for generating genomic databases. JBrowse can handle data in a variety of formats, and is relatively easy to install on a Linux- or Mac OS X-based web server (Skinner and Holmes 2010). JBrowse browsers support plant genomes (e.g. Phytozome), animal genomes (e.g. the Rat Genome Database), and disease-related databases of human data (e.g. the COSMIC Genome Browser).

An example of using JBrowse to view a customized genome assembly and associated annotations is at the *Mnemiopsis* Genome Project (MGP) Portal at the National Human Genome Research Institute (NHGRI) of the US National Institutes of Health (NIH). *Mnemiopsis leidyi* is a type of ctenophore, or comb jelly, a phylum of gelatinous zooplankton found in all the world's seas. The members of this phylum are called comb jellies because of their highly ciliated comb rows, providing their primary means of locomotion, and these early branching metazoans have proven to be an important model organism for understanding the diversity and complexity seen in the early evolution of animals. The *Mnemiopsis* data featured in this portal are the first set of whole genome sequencing data on any ctenophore species to be published and made available to the scientific community (Moreland et al. 2014). The portal provides not only genomic and protein model sequence data, but also a BLAST search interface, pathway and protein domain analysis, and a customized genome browser, implemented in JBrowse, to display the annotation data.

The *Mnemiopsis* genome was assembled into 5100 scaffolds using next generation sequence data from the Roche 454 and Illumina GA-II methods of sequencing (Ryan et al. 2013). The *Mnemiopsis* protein-coding gene models were predicted by integrating the results of ab initio gene prediction programs with RNA-seq transcript data and sequence similarity to other protein datasets. A view of one of those scaffolds is shown in Figure 4.23. As with the UCSC and Ensembl Genome Browsers, data are organized in horizontal tracks, and exons are shown as colored boxes. The first track, *SCF*, is the scaffold. The gene model track, labeled 2.2, displays the exons of the predicted gene models. The next track, called *PFAM2.2*, highlights Pfam domains found in the gene model. The *Mnemiopsis* RNA-seq reads were assembled into transcripts using the Cufflinks program (Trapnell et al. 2010), and the *CL2* track shows the alignment of those transcripts to the genomic scaffold. The *MASK* track highlights repetitive regions. The *EST* and *GBNT* tracks show, respectively, the alignment of publicly available *Mnemiopsis* EST and other RNA sequences from GenBank. These two tracks are empty in this region, so the gene in the gene model track is a novel gene prediction. The overlap between the exons on the Pfam and gene model tracks shows that the predicted gene contains known protein domains. The *CL2* track lends further support to the gene prediction, as the exons of the experimentally derived *Mnemiopsis* transcripts overlap the exons on the gene model track.

Navigation in JBrowse is fairly straightforward, especially for those already accustomed to using the UCSC or Ensembl Genome Browsers. Tracks can be added or removed from display by using the checkboxes on the left side of the window. On the display window, click on a track name and drag it to move the track up or down. To shift the focus of the display window upstream or downstream, click on the display and drag it to the left or right. The left and right arrows at the top of the page also move the display window. JBrowse provides multiple ways to zoom in and out. One option is to use the plus and minus magnifying glasses at the top of the page. Alternatively, place the mouse in the sequence coordinates above the top track and click and drag to highlight a region and zoom in on it. Double clicking on a region also zooms in. Clicking on a track feature opens a window with additional information about that feature. For example, on the MGP Portal, clicking on a gene model in the 2.2 track opens the Gene Wiki

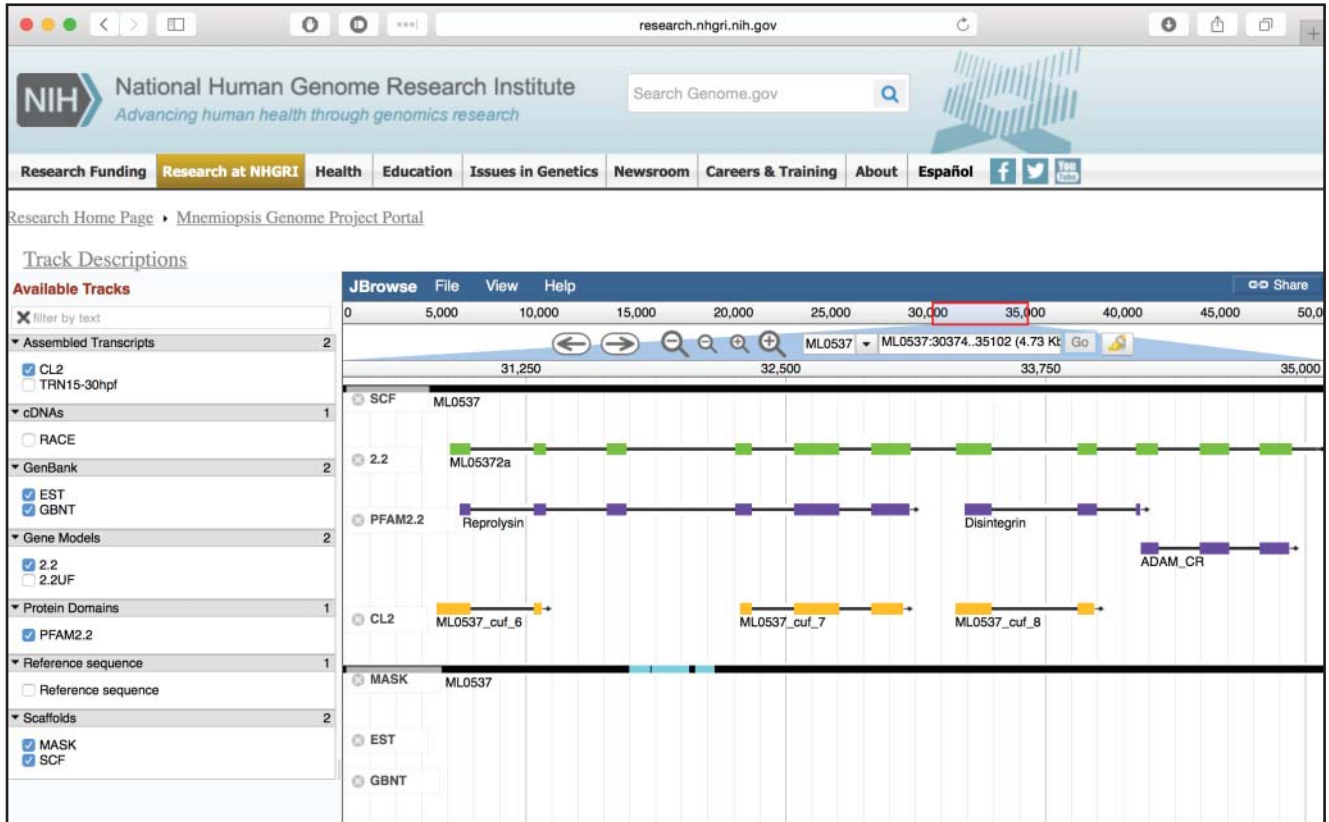


Figure 4.23 JBrowse display of a predicted *Mnemiopsis* gene (*ML05372a*) from the *Mnemiopsis* Genome Project Portal at the National Human Genome Research Institute. Seven tracks are shown on this display: *SCF*, assembled genomic regions are solid black and intermittent gaps are shaded bright pink; *2.2*, consensus *Mnemiopsis* gene models; *PFAM2.2*, non-redundant *Mnemiopsis* protein domains derived from Pfam; *CL2*, RNA-seq reads derived from *Mnemiopsis* embryos, assembled into transcripts using Cufflinks (Trapnell et al. 2010); *MASK*, genomic regions that have been repeat-masked using VMatch are shaded in light blue; *EST*, *Mnemiopsis* expressed sequence tags (ESTs) from GenBank; *GBNT*, *Mnemiopsis* mRNAs and other non-EST RNAs from GenBank.

for that model, a detailed page that includes nucleotide and protein sequences, pre-computed BLAST searches, and annotated Pfam domains. Note that although the general look and feel of JBrowse will remain similar across different genomes, individual JBrowse developers will create tracks and customizations that are specific to their genome project.

Summary

The UCSC and Ensembl Genome Browsers are sophisticated tools that provide free, web-based access to genome assemblies and annotations. This chapter has focused on examples from the human genome and a subset of the annotation tracks available for it. By adding tracks to the default view, users are able to view annotated genes, sequence variants, gene regulatory regions, gene expression data, and much more. The displays are highly customizable, and users can choose which data to view, the display style, and, in some cases, even change the colors of the annotated features. Both browsers can be accessed not only by text-based queries, such as gene symbol or chromosomal position, but also by searches with either nucleotide or protein sequences. The UCSC Genome Browser supports the BLAT search engine, while Ensembl supports both BLAT and BLAST, depending on the analysis type. Furthermore, the UCSC Table Browser and Ensembl's BioMart provide alternate entry points into the underlying data at each site, in which queries can be constructed using a web-based interface and data returned as text that can be downloaded and further manipulated. Although the examples illustrated in this

chapter all derive from the GRCh38 assembly of the human genome, both UCSC and Ensembl host assemblies from many other organisms. The genomes may be assembled in shorter scaffolds, rather than chromosomes, and the variety of annotation types may be much smaller, but the basic look and feel of the genome browser will remain the same across different species.

With new developments in sequencing technology, even smaller laboratories are now able to generate whole genome sequencing data, including ChIP-Seq and RNA-seq, exome and genome sequencing, and even novel genome assemblies. Starting in 2015, genomic data sharing policies now require that all NIH-funded research that generates large-scale genomic data be submitted to a public database in a timely manner. While human data must be submitted to an NIH-designated data repository, as of this writing, non-human data may be made available through any widely used data repository. Viewing and sharing these data with the larger community of biologists may best be done with a genome browser. Both the UCSC and Ensembl Genome Browsers provide the option for users to upload their own annotations and view them in the context of the public genome data. Using *Sessions* or *Track Hubs*, users can share these data with colleagues. The *Assembly Hubs* feature at UCSC now allows users to share novel genomes using the Genome Browser framework. Furthermore, the source code for the UCSC Genome Browser is publicly available, so others are free to set up their own browsers to host their own annotations, or even their own genomes. Alternatively, researchers who want to host their own genome browser should consider JBrowse. This freely available software tool can be easily installed on a web server and used to host custom genomes and annotations.

The UCSC and Ensembl teams start with the same source of data, a genome assembly, often provided by the GRC. Each team then layers on its own annotations from different sources, including the location of genes, from GENCODE, RefSeq, and other gene prediction pipelines, and variants, from NCBI's dbSNP. Both browsers also include the location of experimentally determined epigenetic marks, including histone modifications, as well as DNaseI hypersensitive sites, both of which can inform predictions of gene regulatory regions. The regulatory tracks at UCSC come from the ENCODE project, while Ensembl provides a *Regulatory Build*, which includes data from ENCODE as well as other sources. Although individual researchers may have personal preferences about which interface is easier to use, or which site provides information that is more relevant to the biological question they are studying, most members of the bioinformatics community will undoubtedly use a genome browser at some point in their research career.

Internet Resources

UCSC Genome Browser

Main page	genome.ucsc.edu
Genome Browser User's Guide	genome.ucsc.edu/goldenPath/help/hgTracksHelp.html
Table Browser User's Guide	genome.ucsc.edu/goldenPath/help/hgTablesHelp.html
Displaying custom annotation data	genome.ucsc.edu/goldenPath/help/customTrack.html
Data file formats for custom annotation	genome.ucsc.edu/FAQ/FAQformat.html
Sessions User's Guide	genome.ucsc.edu/goldenPath/help/hgSessionHelp.html
Using UCSC Genome Browser Track Hubs	genome.ucsc.edu/goldenPath/help/hgTrackHubHelp.html
Assembly Hubs wiki	genomewiki.ucsc.edu/index.php/Assembly_Hubs
Contact information	genome.ucsc.edu/contacts.html

Ensembl Genome Browser

Main page	www.ensembl.org
Ensembl Stable IDs	www.ensembl.org/info/genome/stable_ids
Ensembl Archives	www.ensembl.org/info/website/archives
BioMart	www.ensembl.org/biomart/martview

pre!Ensembl	pre.ensembl.org
Help & Documentation	www.ensembl.org/info
BioMart documentation	www.ensembl.org/info/data/biomart
Displaying custom annotation data	www.ensembl.org/info/website/upload
Data file formats for custom annotation	www.ensembl.org/info/website/upload/index.html#formats
Contact information	www.ensembl.org/info/about/contact
JBrowse	
JBrowse Genome Browser	jbrowse.org
COSMIC Genome Browser	cancer.sanger.ac.uk/cosmic/browse/genome
<i>Mnemiopsis</i>	research.nhgri.nih.gov/mnemiopsis
Genome Project Portal (MGAP)	
Phytozome	phytozome.jgi.doe.gov
Rat Genome Database	rgd.mcw.edu
Other genome resources	
GENCODE	www.genecodegenes.org
Genome Reference Consortium	www.ncbi.nlm.nih.gov/grc
GWAS Catalog	www.ebi.ac.uk/gwas
National Center for Biotechnology Information (NCBI) Genome Data Viewer	www.ncbi.nlm.nih.gov/genome/gdv
National Institutes of Health (NIH) Genomic Data Sharing Policyies	osp.od.nih.gov/scientific-sharing/policies
NIH Genotype-Tissue Expression (GTEx) Portal	www.gtexportal.org
Track Hub Registry	www.trackhubregistry.org

Further Reading

The best way to learn about the data and tools available in the UCSC and Ensembl Genome Browsers is to read the relevant sections of the online documentation that accompanies each browser. The documentation on both sites is extensive and up to date, and will likely answer the user's questions. Alternatively, specialized questions can be addressed by contacting the web site development teams. URLs are listed in the Internet Resources section of this chapter.

The Database issue of *Nucleic Acids Research*, published in January of each year, usually includes articles that provide a broad overview of each Genome Browser as well as a description of new data and resources. The references for 2019 are listed below, and additional information can be found in Chapter 1.

References

- Aken, B.L., Ayling, S., Barrell, D. et al. (2016). The Ensembl gene annotation system. *Database (Oxford)* 2016, pii: baw093.
- Benson, D.A., Boguski, M.S., Lipman, D.J., and Ostell, J. (1997). GenBank. *Nucleic Acids Res.* 25 (1): 1–6.
- Buels, R., Yao, E., Diesh, C.M. et al. (2016). JBrowse: a dynamic web platform for genome visualization and analysis. *Genome Biol.* 17: 66.
- Buniello, A., MacArthur, J.A.L., Cerezo, M. et al. (2019). The NHGRI-EBI GWAS Catalog of published genome-wide association studies, targeted arrays and summary statistics. *Nucleic Acids Res.* 47 (D1): D1005–D1012.

- Camacho, C., Coulouris, G., Avagyan, V. et al. (2009). BLAST+: architecture and applications. *BMC Bioinformatics* 10: 421.
- Cunningham, F., Achuthan, P., Akanni, W. et al. (2019). Ensembl 2019. *Nucleic Acids Res* 47 (D1): D745–D751.
- ENCODE Project Consortium (2012). An integrated encyclopedia of DNA elements in the human genome. *Nature* 489 (7414): 57–74.
- Frankish, A., Uszczynska, B., Ritchie, G.R. et al. (2015). Comparison of GENCODE and RefSeq gene annotation and the impact of reference geneset on variant effect prediction. *BMC Genomics* 16 (Suppl 8): S2.
- Goffeau, A., Barrell, B.G., Bussey, H. et al. (1996). Life with 6000 genes. *Science* 274 (5287): 546, 563–567.
- Green, E.D. (2001). Strategies for the systematic sequencing of complex genomes. *Nat. Rev. Genet.* 2 (8): 573–583.
- GTEX Consortium (2015). Human genomics. The Genotype-Tissue Expression (GTEx) pilot analysis: multitissue gene regulation in humans. *Science* 348 (6235): 648–660.
- Haeussler, M., Zweig, A.S., Tyner, C. et al. (2019). The UCSC Genome Browser database: 2019 update. *Nucleic Acids Res* 47 (D1): D853–D858.
- Harrow, J., Frankish, A., Gonzalez, J.M. et al. (2012). GENCODE: the reference human genome annotation for The ENCODE Project. *Genome Res.* 22 (9): 1760–1774.
- Herrero, J., Muffato, M., Beal, K. et al. (2016). Ensembl comparative genomics resources. *Database (Oxford)* 2016: bav096.
- Howald, C., Tanzer, A., Chrast, J. et al. (2012). Combining RT-PCR-seq and RNA-seq to catalog all genic elements encoded in the human genome. *Genome Res.* 22 (9): 1698–1710.
- Hubbard, T., Barker, D., Birney, E. et al. (2002). The Ensembl genome database project. *Nucleic Acids Res.* 30 (1): 38–41.
- Kent, W.J. (2002). BLAT—the BLAST-like alignment tool. *Genome Res.* 12 (4): 656–664.
- Kent, W.J. and Haussler, D. (2001). Assembly of the working draft of the human genome with GigAssembler. *Genome Res.* 11 (9): 1541–1548.
- Kent, W.J. and Zahler, A.M. (2000). The intronator: exploring introns and alternative splicing in *Caenorhabditis elegans*. *Nucleic Acids Res.* 28 (1): 91–93.
- Kinsella, R.J., Kähäri, A., Haider, S. et al. (2011). Ensembl BioMart: a hub for data retrieval across taxonomic space. *Database (Oxford)* 2011: bar030.
- Kitts, P. (2003). Genome assembly and annotation process. In: *The NCBI Handbook* (eds. J. McEntyre and J. Ostell) ch. 14. Bethesda, MD: National Center for Biotechnology Information.
- Lander, E.S., Linton, L.M., Birren, B. et al., and International Human Genome Sequencing Consortium (2001). Initial sequencing and analysis of the human genome. *Nature* 409 (6822): 860–921.
- Lawrence, M., Daujat, S., and Schneider, R. (2016). Lateral thinking: how histone modifications regulate gene expression. *Trends Genet.* 32 (1): 42–56.
- Li, R., Fan, W., Tian, G. et al. (2010). The sequence and de novo assembly of the giant panda genome. *Nature* 463 (7279): 311–317.
- McLaren, W., Gil, L., Hunt, S.E. et al. (2016). The Ensembl variant effect predictor. *Genome Biol.* 17 (1): 122.
- Moreland, R.T., Nguyen, A.D., Ryan, J.F. et al. (2014). A customized Web portal for the genome of the ctenophore *Mnemiopsis leidyi*. *BMC Genomics* 15: 316.
- Mudge, J.M. and Harrow, J. (2015). Creating reference gene annotation for the mouse C57BL6/J genome assembly. *Mamm. Genome* 26 (9–10): 366–378.
- Ryan, J.F., Pang, K., Schnitzler, C.E. et al. (2013). The genome of the ctenophore *Mnemiopsis leidyi* and its implications for cell type evolution. *Science* 342 (6164): 1242592.
- Sayers, E.W., Agarwala, R., Bolton, E.E. et al. (2019). Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res* 47 (D1): D23–D28.
- Skinner, M.E. and Holmes, I.H. (2010). Setting up the JBrowse genome browser. *Curr. Protoc. Bioinformatics*. Chapter 9: Unit 9.13. <https://doi.org/10.1002/0471250953.bi0913s32>.

- Trapnell, C., Williams, B.A., Pertea, G. et al. (2010). Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat. Biotechnol.* 28 (5): 511–515.
- Wheeler, D.L., Church, D.M., Lash, A.E. et al. (2001). Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.* 29 (1): 11–16.
- Wolfsberg, T.G., Wetterstrand, K.A., Guyer, M.S. et al. (2002). A user's guide to the human genome. *Nat. Genet.* 32 (Suppl): 1–79.
- Wu, P.Y., Phan, J.H., and Wang, M.D. (2013). Assessing the impact of human genome annotation choice on RNA-seq expression estimates. *BMC Bioinformatics* 14 (Suppl 11): S8.
- Zerbino, D.R., Johnson, N., Juetteman, T. et al. (2016). Ensembl regulation resources. *Database (Oxford)* 2016, pii: bav119.

This chapter was written by Dr. Tyra G. Wolfsberg in her private capacity. No official support or endorsement by the National Institutes of Health or the United States Department of Health and Human Services is intended or should be inferred.

5

Genome Annotation

David S. Wishart

Introduction

Thanks to rapid advances in DNA sequencing technology and DNA analysis software, genome projects that used to take years and cost millions of dollars to finish can now be completed in just weeks at a cost of a few thousand dollars. The typical workflow for a modern genome sequencing project involves performing whole genome DNA sequencing of selected organisms using a next generation DNA sequencer, running a variety of programs to assemble a reference genome, and using software to locate and identify all of the protein-coding ribosomal RNA (rRNA) and transfer RNA (tRNA) genes within the genomic sequence. This last process is called *genome annotation* and it is the primary subject of this chapter. Strictly speaking, genome annotation is not genome prediction. Gene or genome prediction is a subfield of genome annotation. In particular, gene prediction uses mathematical or probabilistic models to analyze DNA sequences and to identify gene boundaries and gene structures. On the other hand, genome annotation uses gene (and genome) prediction results along with other lines of evidence such as gene expression data, protein expression data, sequence homology to other annotated genomes, and even literature assessments to generate a set of genome annotations. These annotations include not only the location of the genes on each chromosome but also their names (based on homology), calculated properties (such as sequence length, amino acid composition, and molecular weight), expression levels (if available), and probable functions.

Depending on the type of organism that has been sequenced, the task of genome annotation can be either quite easy or quite difficult. Prokaryotes (including bacteria and archaea) have relatively small genomes, typically no more than 5 million base pairs, consisting of one or two circular chromosomes and perhaps one or two small plasmids. The gene structure for prokaryotes is very simple, with each gene being a contiguous open reading frame (ORF). Furthermore, the coding density for prokaryotes is very high, with at least 85–90% of their DNA coding for proteins, tRNAs, and rRNAs (Hou and Lin 2009). This makes the identification of genes in prokaryotes relatively simple. On the other hand, eukaryotic gene identification is often quite difficult. This is because eukaryotes have very large genomes (often billions of base pairs), with very low (often <2%) coding densities (Hou and Lin 2009). Eukaryotic gene structure is also much more complex than prokaryotic gene structure. In particular, eukaryotic genes are split into exons and introns, and most eukaryotic genes are separated by very large stretches of non-coding DNA (called intergenic regions).

While the cellular machinery in eukaryotic cells is able to recognize and process gene signals with remarkable accuracy and precision, our understanding of the molecular mechanisms by which eukaryotic sequence signals are recognized and processed remains incomplete. As a result, currently available eukaryotic gene prediction methods are not very accurate. Therefore, in the absence of additional experimental or extrinsic information (e.g. gene expression data), one should assume that eukaryotic gene predictions are only approximate. Even with considerable experimental data at hand, it is still quite difficult to fully annotate the best-studied

eukaryotic genomes. For instance, the DNA sequence for the human genome has been known since 2001, but the actual number of genes encoded within our own genome has still not been fully determined (Pennisi 2003; Ezkurdia et al. 2014).

This chapter briefly reviews some of the computational methods and algorithms underlying computational gene prediction for both prokaryotes and eukaryotes. It also describes how experimental evidence and database comparisons can be integrated into these gene prediction tools to improve gene prediction performance and to ensure more complete genome annotation. Methods for assessing the performance of computational gene finders are also described. Finally, a number of genome annotation pipelines are highlighted, along with several tools for visualizing the resulting annotations.

Gene Prediction Methods

Different methods for gene prediction have been developed separately for prokaryotes and eukaryotes because of important differences in their overall gene organization. Gene-finding programs, whether for prokaryotes or eukaryotes, fall into two general categories: *intrinsic* (or *ab initio*) gene predictors and *extrinsic* (or evidence based) gene finders (Borodovsky et al. 1994).

Ab initio gene prediction approaches attempt to predict and annotate genes solely using DNA sequence data as input and without direct comparison with other sequences or sequence databases. *Ab initio* approaches involve searching for sequence signals that are potentially involved in gene specification and/or looking for regions that show compositional bias that has been correlated with coding regions. This combined approach to gene finding is called searching by signal and searching by content. GeneMark (Borodovsky and McIninch 1993), GLIMMER (Delcher et al. 1999, 2007), EasyGene (Larsen and Krogh 2003), and GENSCAN (Burge and Karlin 1997) are well-known examples of intrinsic or *ab initio* gene-finding programs. In contrast, extrinsic gene-finding methods involve both homology-based and comparative approaches, in which the gene structure is determined through comparison with other sequences whose characteristics are already known. BLASTX is an example of an extrinsic gene-finding program that has been frequently applied for gene identification in prokaryotic genomes (Borodovsky et al. 1994). Extrinsic gene prediction methods depend on having experimental evidence (such as messenger RNA (mRNA) or RNA-seq data) and/or a large body of pre-existing experimental sequencing data to perform sequence comparisons and gene identifications. We will discuss these extrinsic methods and their role in genome annotation a little later in this chapter. To begin with, we will focus on the intrinsic or *ab initio* gene prediction methods.

Ab Initio Gene Prediction in Prokaryotic Genomes

A prokaryotic gene typically begins with a start codon (e.g. ATG), ends with one of three stop codons (e.g. TAG, TAA, or TGA), and is usually at least 100 bases long (Figure 5.1). These protein-coding genes are called ORFs. Most of the genes in prokaryotic genomes are organized into *operons*, which are gene clusters consisting of more than one ORF that are under the control of a shared set of regulatory sequences. These regulatory sequences can include enhancers, silencers, terminators, operators, or promoters. Regulatory sequences typically constitute the 10–15% of the prokaryotic genome that is not coding for protein sequences. A prokaryotic gene promoter is a small segment of DNA that initiates transcription of a particular gene. Promoters are located near the transcription start sites (TSSs) of genes, on the same strand and upstream of the gene or ORF. In prokaryotes, the promoter contains two short sequence elements approximately 10 bases and 35 nucleotides upstream from the TSS. The element located 10 bases upstream is called the TATA box in archaea or the Pribnow (TATAAT) box in bacteria (Pribnow 1975). These abbreviations or letters actually indicate the consensus DNA sequences

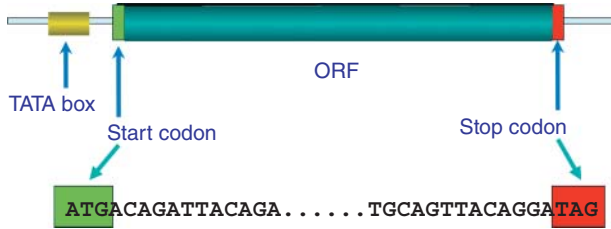


Figure 5.1 A simplified depiction of a prokaryotic gene or open reading frame (ORF) including the start codon (or translation initiation site), the stop codon (TAG), and the TATA or Pribnow box.

seen for these regions. In addition to the TSS, almost all prokaryotic genes have a ribosome binding site (RBS) that is 8–10 bases upstream of the start (ATG) codon. The start codon is also called the translation initiation site (TIS). The RBS exhibits a specific nucleotide pattern (AGGAGG) called a Shine–Dalgarno (SD) consensus sequence (Shine and Dalgarno 1975). The SD sequence enables interactions between mRNA and the cell’s translational machinery. In bacteria and archaea, translation initiation is generally thought to occur through the base-pairing interaction between the 3’ tail of the 16S rRNA of the 30S ribosomal subunit and the site in the 5’ untranslated region (UTR) of an mRNA that carries the SD consensus.

Consensus sequences, while providing a useful reminder or mnemonic, are never really used in modern gene signal or gene site (i.e. TIS, RBS, TSS, and terminator) identification. Instead, most gene signals can be identified by using positional weight matrices (PWMs) or position-specific scoring matrices (PSSMs; see also Chapter 3). These scoring matrices are calculated from carefully aligning a set of known functional signals and determining the adjusted frequency with which specific bases may appear in certain positions. An example of how to calculate a PSSM is given in Box 5.1. Once calculated for a given signal, signal-specific PSSMs can be used to rapidly compute, along the length of a sequence of interest, the position and likelihood of the selected gene signals. A simplified gene prediction protocol for prokaryotes involves the following steps.

- Start at the beginning of the genome sequence at the 5’ end of one DNA strand and find an ATG start codon that makes the longest ORF (minimum 150 bases), then move to the next ATG downstream of the previously identified ORF and repeat the process for the rest of the genome sequence.
- Repeat the above process for the opposite DNA strand.
- For all identified ORFs, score the quality of the TSS and RBS signals using site-specific PSSMs to refine the ORF predictions and produce a final list of genes.

Box 5.1 Position-Specific Scoring Matrices

Position-specific scoring matrices (PSSMs), which are also called positional weight matrices (PWMs) or positional specific weight matrices (PSWMs), are usually derived from a set of aligned sequences that are believed to be functionally related. In this example, five different DNA sequences consisting of 10 bases each, which are believed to be functionally related (as promoter regions), are aligned.

```

A T T T A G T A T C
G T T C T G T A A C
A T T T T G T A G C
A A G C T G T A A C
C A T T T G T A C A

```

From this alignment, a simple positional frequency matrix (PFM) can be generated. In this matrix the frequency of the As, Cs, Gs, and Ts is tabulated (based on the above alignment) for each of the 10 base positions. So in the first position there are three As, one C, one G, and no Ts (see column 1). The PFM for the above alignment is:

(Continued)

Box 5.1 (Continued)

```

A 3 2 0 0 1 0 0 5 2 1
C 1 0 0 2 0 0 0 0 1 4
G 1 0 1 0 0 5 0 0 1 0
T 0 3 4 3 4 0 5 0 1 0

```

The PFM can now be converted to a positional probability matrix (PPM). A PPM is a matrix consisting of a set of decimal values based on the percentage or frequency of occurrences of each base in each position in the sequence alignment. In other words, we must normalize the frequencies by dividing the nucleotide count at each position by the number of sequences in the alignment. So if there are five sequences in the alignment and three As in the first position, then the positional probability for A in the first position is $3/5 = 0.6$. Likewise, if there is one C in the first position, its positional probability is $1/5 = 0.2$. One G corresponds to a positional probability of 0.2, and no Ts corresponds to a positional probability of 0 (see column 1). Performing this same calculation across all 10 positions of the alignment, the full PPM would appear as follows:

```

A .6 .4 0 0 .2 0 0 1 .4 .2
C .2 0 0 .4 0 0 0 0 .2 .8
G .2 0 .2 0 0 1 0 0 .2 0
T 0 .6 .8 .6 .8 0 1 0 .2 0

```

The probabilities in the above PPM can be multiplied together to calculate the probability that a given DNA sequence is closely related to the original five sequences. For instance, if we wanted to know if the new sequence ATTTTGTATA is closely related, we could multiply the values for each sequence position to calculate that sequence's probability:

$$p = 0.6 \times 0.6 \times 0.8 \times 0.6 \times 0.8 \times 1 \times 1 \times 1 \times 0.2 \times 0.2 = 0.0055$$

Note that if we had performed this same calculation on an almost identical sequence such as ACTTTGTATA (which differs by only one base) we would get $p = 0$. We get a 0 probability because C was not observed in the second position of our training set. Building a PPM with only five sequences means you are very likely to underestimate (or overestimate) the true fractional frequencies of each base, leading to problems in calculating probabilities similar to what we just saw. To account for the small size of our multiple sequence alignment (MSA) we should introduce *pseudocounts*. Pseudocounts are used to avoid issues that result from matrix entries having a value of 0. Pseudocounting is equivalent to multiplying each column of the PPM by a Dirichlet distribution, thereby allowing the probability to be calculated for the "unseen" or unused sequences. A simple way of doing this is to normalize the data to match the overall base composition of the genome(s) being considered and to add a correction factor that scales as the square root of the number of sequences in the MSA. Hence, the following formula can be used to rescore each base position in the PPM:

$$\text{score}(X_i) = (Q_x + P_x)/(N + B)$$

where Q_x is the number of counts of base type X at position i , P_x is the number of pseudocounts of base type X , which is equal to $B \times$ the frequency of base type X , N is the total number of sequences in the MSA, and B is the number of pseudocounts (assumed to be \sqrt{N}). For the genome or genomes of interest the frequency of As is 0.32, Ts is 0.32, Cs is 0.18, and Gs is 0.18. Using this information the value for A in the first position is $(3 + (\sqrt{5} \times 0.32))/(5 + \sqrt{5}) = 0.51$. The value for C in the second position is $(1 + (\sqrt{5} \times 0.18))/(5 + \sqrt{5}) = 0.19$, and so on. The pseudocount corrected PPM is now:

```

A .51 .38 .09 .09 .24 .09 .09 .79 .38 .24
C .19 .06 .06 .33 .06 .06 .06 .06 .19 .61
G .19 .06 .19 .06 .06 .75 .06 .06 .19 .06
T .09 .51 .65 .51 .65 .09 .79 .09 .24 .09

```

Ideally each of the columns should sum to 1 but, because of rounding, the sums in this example are sometimes slightly above or below 1. With this rescored matrix, you will notice that there are now no zero entries. However, the calculation of probabilities through multiplication is tedious (given the number of significant digits) and difficult. A simpler way is to convert the PPM to a different type of matrix by taking the negative \log_{10} of each number in the PPM. This converts two-digit decimals to single-digit decimals and it also allows one to add rather than multiply to calculate probabilities. If we take the $-\log_{10}$ of the above PPM, we get:

```
A 0.3 0.4 1.0 1.0 0.6 1.0 1.0 0.1 0.4 0.6
C 0.7 1.2 1.2 0.5 1.2 1.2 1.2 1.2 0.7 0.2
G 0.7 1.2 0.7 1.2 1.2 0.1 1.2 1.2 0.7 1.2
T 1.1 0.3 0.2 0.3 0.2 1.0 0.1 1.0 0.6 1.0
```

This modified matrix is called a *log likelihood scoring matrix* or a PSSM. Using the above PSSM, we can now calculate the score (or the log likelihood) for the query sequence ATTTGTATA: $0.3 + 0.3 + 0.2 + 0.3 + 0.2 + 0.1 + 0.1 + 0.1 + 0.6 + 0.6 = 2.8$. The sequence score gives an indication of how different the sequence is from a random sequence. The higher the score, the more likely the sequence is a promoter/functional site and not a random sequence. A score of 2.8 is very high. The sequence score can also be interpreted in terms of the binding energy for that sequence.

However, such a simplified algorithm would only likely be 75–80% correct (Besemer et al. 2001). This is because prokaryotic genes are not always so simple to identify. For instance, the ATG start codon is not always used for all bacterial genes. Among the 4284 genes identified in *Escherichia coli*, 83% use ATG, 14% use GTG and 3% use TTG start codons (Blattner et al. 1997). Likewise, using a simple rule to identify only long ORFs may miss many short ORFs or misidentify ORFs that have an unusual codon bias (indicating they are unlikely to code for a gene). Indeed, the length distributions of ORFs known to code for proteins compared with ORFs that occur by chance differ quite significantly. More specifically, coding ORFs have a length distribution that resembles the gamma distribution (see Glossary), while non-coding ORFs have a length distribution that resembles a simple exponential function (Lukashin and Borodovsky 1998). In addition to these complications, it has recently been found that certain prokaryotic genes have very unusual gene start signals because of a phenomenon called *leaderless transcription* (Slupska et al. 2001). In leaderless transcription, RNA transcripts have very short 5' UTRs, with a length < 6 bases. These regions are so short that they are unable to host the RBS. This places the TSS at or very near to the TIS. In these cases, the promoter signal has to be used for more accurate TIS identification.

Given the variations in the length and character of many prokaryotic gene signals, PSSMs are not the most effective signal recognition tools available. More advanced methods of gene signal recognition exist, such as Markov models (Box 5.2), hidden Markov models or HMMs (Box 5.3), artificial neural networks, and support vector machines. These machine learning methods do a far better job of handling variable lengths and conditional sequence dependencies that cannot be captured with simple PSSMs.

Box 5.2 Markov Models

A Markov chain, model, or process refers to a series of observations in which the probability of an observation depends on a number of previous observations. The number of observations defines the “order” of the chain. For example, in a first-order Markov model, the probability of an observation depends only on the previous observation. In a Markov chain of order 5, the probability of an observation depends on the five preceding observations. A DNA sequence can be considered to be an example of a Markov model because the likelihood of observing a particular base at a given position may depend on the bases

(Continued)

Box 5.2 (Continued)

preceding it. In particular, in coding regions, it is well known that the probability of a given base depends on the five preceding bases, reflecting observed codon biases and dependencies between adjacent codons. In non-coding regions, such dependence is not observed. When scanning an anonymous genomic region, one can compute how well the local nucleotide sequence conforms to the fifth-order dependencies observed in coding regions and assign appropriate coding likelihood scores.

Box 5.3 Hidden Markov Models in Gene Prediction

Hidden Markov models (HMMs) are used to provide a statistical representation of real biological processes. They have found widespread use in many areas of bioinformatics, including multiple sequence alignment, the characterization and classification of protein families, the comparison of protein structures, and the prediction of gene structure.

In this chapter, all of the gene-finding methods that are described have two things in common: they use a raw nucleotide sequence as their input and, for each position in the sequence, they attempt to predict whether a given base is most likely found in an intron, an exon, or within an intergenic region. In making these predictions, the algorithm applied (HMM or otherwise) must take into account what is known about the structure of a gene, showed in a simplified fashion in Figure 5.2.

Working from the 5' to 3' end of the gene, the method must take into account the unique characteristics of promoter regions, transcription start sites, 5' UTRs, start codons, exons, splice donors, introns, splice acceptors, stop codons, 3' UTRs, and polyA tails. In addition to any conserved sequences or compositional bias that may characterize each of these regions (Box 5.1), the method also needs to take into account that each of these elements appears with a controlled syntax; for example, the promoter (and its TATA box) must appear before the start codon, an initial exon must follow the start codon, introns must follow exons, introns can only be followed by internal or terminal exons, stop codons cannot interrupt the coding region, and polyA signals must appear after the stop codon. Finally, an ORF must be maintained throughout to produce a protein once all is said and done.

Each of the elements – exons, introns, and so forth – are referred to as *states*. The sequence characteristics and syntactical constraints described above allow a transition probability to be assigned, indicating how likely a change of state is as one moves through

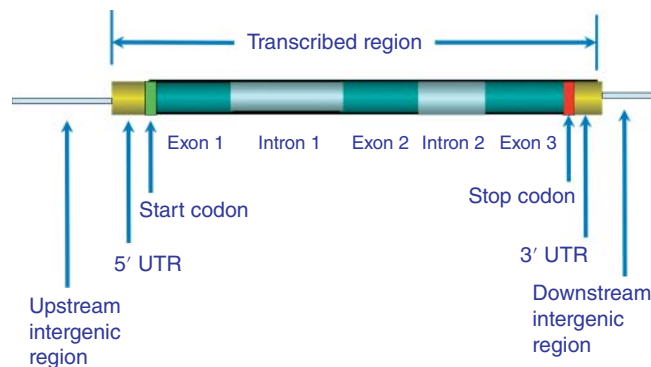


Figure 5.2 A simplified depiction of a eukaryotic gene illustrating the multi-intron/exon structure, the location of the start and stop codons, the untranslated regions (UTRs), and the intergenic regions that surround the transcribed gene.

the sequence, base by base. Although the user “sees” the nucleotide sequence being analyzed, the user does not actually “see” the states that the individual bases are in – hence the term *hidden*. Put otherwise, each state emits a particular kind of nucleotide sequence, with its own emission probability; the state emitting the nucleotide sequence is hidden, but the sequence itself is visible. The transition and emission probabilities are derived from training sets, sequences for which the correct gene structure is already known. The goal here is to develop a set of parameters that allow the method to be fine-tuned, maximizing the chances that a correct prediction is generated on a new sequence of interest. As alluded to in the text, these parameters differ from organism to organism, and the success of any given HMM-based method depends on how well these parameters have been deduced from the training set.

The most advanced ab initio prokaryotic gene-finding programs incorporate all of the above-mentioned caveats and conditions. In particular, they handle alternate start codons, they accommodate differential codon bias, they model gene length distributions, they calculate typical and atypical GC content, they deal with leaderless transcription, and they incorporate machine learning-based or advanced signal recognition techniques to identify key gene signals. GeneMarkS (Borodovsky and Lomsadze 2011), GLIMMER (Delcher et al. 2007), EasyGene (Larsen and Krogh 2003), and Prodigal (Hyatt et al. 2010) are all examples of very advanced prokaryotic gene-finding programs. They are remarkably accurate (>97% on average) at detecting validated protein-coding ORFs (Besemer et al. 2001; Delcher et al. 2007; Hyatt et al. 2010). However, to get this kind of performance these prokaryotic gene-finding programs need to be trained on well-annotated genomes from bacterial species that are already similar in sequence to the genome being analyzed. If the programs are not specifically trained, they do not do quite as well (90–95% accuracy).

Ab Initio Gene Prediction in Eukaryotic Genomes

A diagram of how eukaryotic genes are organized is shown in Figure 5.2. As can be seen from this figure, eukaryotic genes are somewhat more complex than prokaryotic genes. In particular, the density of protein-coding regions for eukaryotic genomes (and especially vertebrate genomes) is 90–100 times lower than it is for prokaryotic genomes. These sparse protein-coding regions are separated by long stretches of intergenic DNA while their coding sequences (the exons) are interrupted by large, non-coding introns. Genes are recognized and transcribed by eukaryotic RNA polymerases, and the resulting long RNA transcripts are then cut by various small ribonuclear proteins (snRNPs) to remove the introns (Will and Lührmann 2011). The remaining exons are then spliced together to form the much smaller protein-coding transcript. The snRNPs recognize specific cut sites at the exon/intron junctions to ensure that the splicing is always performed precisely.

In the human genome, just 1.1% of the genome is composed of exons, 24% is composed of introns, and 75% of the genome constitutes intergenic DNA. On average there are 5.48 exons per gene with each exon encoding a peptide fragment of 30–36 amino acids (Sakharkar et al. 2002). The longest exon in the human genome is 11 555 bases, while the shortest is just two bases long (Sakharkar et al. 2002). Not only are exons “rare,” they vary tremendously in length. What is more, they can be alternately spliced to produce very different combinations of final gene (transcript) products. This makes gene prediction significantly more difficult for eukaryotes than for prokaryotes.

Computational gene prediction for eukaryotes essentially involves mimicking the biological transcriptional and splicing process. In the biological process, various proteins and protein complexes within the cell scan through the DNA sequence, recognize and bind to specific DNA

sites, transcribe the gene, and then cut and splice the transcript to form a final gene product. In the computational process, the proteins are replaced with various algorithms that:

- identify and score suitable splice sites and start and stop signals along the query sequence
- determine the location of the candidate exons, as deduced through the detection of these signals
- score and identify the best exons as a function of both the signals used to detect the exons as well as the coding statistics computed on the putative exon sequence itself
- assemble (or “splice”) a subset of these exon candidates into a predicted gene structure. The assembly is produced in a way that maximizes a particular scoring function that is dependent on the score of each of the individual exon candidates.

The way in which each of these tasks is actually implemented varies from program to program. Rather than discuss each program in detail, we will describe the three major processes common to almost all *ab initio* eukaryotic gene prediction programs: predicting exon-defining signals, predicting and scoring exons, and, finally, exon assembly.

Predicting Exon-Defining Signals

Just as prokaryotic genes have DNA signals, eukaryotic genes have distinct DNA signals as well. Some of these elements are similar to prokaryotes, while others are quite different (Figure 5.3). For instance, many eukaryotic genes have promoter elements that also share some sequence similarity to prokaryotic genes. The most extensively studied core promoter element in eukaryotes is known as the TATA box or the Goldberg–Hogness box (Lifton et al. 1978), found 25–30 base pairs upstream from the TSS. The TATA box is also found in archaea and bacteria and it appears to be a very ancient DNA signal. The TATA box in eukaryotes has the consensus sequence TATA(A/T)A(A/T) and is often coupled to another regulatory sequence called the CCAAT box (consensus: GGCCAATCT), located ~150 base pairs upstream of the TATA box. Only about 25–35% of mammalian genes contain TATA boxes, while the rest contain other kinds of core promoter elements. Eukaryotic genes also contain regulatory sequences beyond the core promoter, including enhancers, silencers, and insulators. These regulatory sequences can be spread over a large genomic distance, often hundreds of kilobases from the core promoters. In addition to having a wide variety of promoter or enhancer signals, eukaryotic genes also have very specific DNA signals to define the location of exons and introns.

More specifically, there are four basic DNA signals involved in defining exons: the TIS, the 5' (or donor) splice site, the 3' (or acceptor) splice site, and the translational stop codon. In eukaryotes, the TIS is defined by the Kozak consensus sequence, often given as ACCATGG (Kozak 1987), where the central ATG is the start codon. The 5' donor splice site is typically defined by a consensus sequence given as GG/GT, while the 3' acceptor splice site has a consensus sequence of CAG/G, where the slash indicates the cut sites for splicing (Figure 5.4). The translational stop codons include the usual TAG, TAA, or TGA.

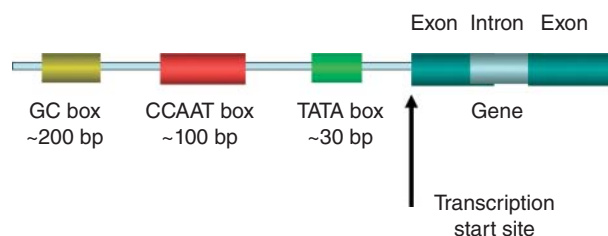


Figure 5.3 A schematic illustration of the upstream regions of a eukaryotic gene with the GC box located ~200 bp upstream, the CCAAT box located ~100 bp upstream, and the TATA box located ~30 bp upstream of the transcription start site.

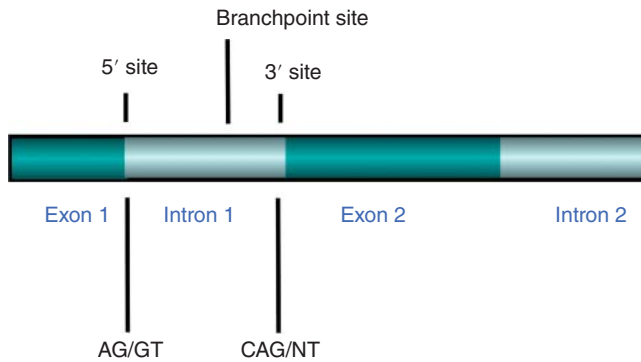


Figure 5.4 A schematic illustration of the splice site regions around exons and introns including the 5' and 3' splice sites and their consensus sequences.

The first methods used to identify exon-defining signals were simple PWMs or PSSMs. These proved to be rather poor at identifying short DNA signals, such as splice sites. As a result, these simple models have since given way to much more advanced pattern recognition techniques such as HMMs (Box 5.3). These powerful pattern recognition approaches allow very complex sequence patterns to be “learned” from large datasets consisting of well-known or well-annotated exon-defining signals. An HMM is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (i.e. hidden) states. HMMs are commonly used in many real-life applications such as speech, handwriting, and gesture recognition. The application of HMMs in bioinformatics began in the early 1990s (Krogh et al. 1994) and led to a significant advance in gene prediction accuracy. HMMs make it possible to define highly complex patterns of variable lengths, including many exon-defining signals such as protein-coding regions (discussed below), donor, acceptor, and lariat sites, as well as translational start and end sites.

Predicting and Scoring Exons

In addition to the identification of exon-defining signals, the accurate prediction of exons also depends on content-based features. Exons can be divided into three basic types:

- *initial exons*: ORFs delimited by a start site and a donor site
- *internal exons*: ORFs delimited by a 5' (donor) site and a 3' (acceptor) site
- *terminal exons*: ORFs delimited by a 3' (acceptor) site and a stop codon.

Most transcribed genes are composed of one initial exon, multiple internal exons, and a single terminal exon. Zhang (2002) provides a more comprehensive discussion of these types of eukaryotic exons.

Exons, by definition, are protein-coding regions. Protein-coding regions are known to exhibit characteristic compositional bias when compared with non-coding regions. These include somewhat richer GC content and a distinctly non-random codon (triplet) frequency preference. The observed codon bias results from the uneven distribution of amino acids in proteins, the uneven use of synonymous codons, and natural selection for translational optimization in coding regions. To discriminate protein-coding regions from non-coding regions, a number of DNA content-based measures were developed in the 1990s (Fickett and Tung 1992; Gelfand 1995; Guigó 1999). These content measures, which are also referred to as *coding statistics*, reflect the likelihood that a given DNA sequence codes for a protein or protein fragment. Many methods for the computation of content-based measures have been published over the years. Some of the first methods measured patterns seen in codon triplet frequencies. However, more information was found in the frequencies of pairs of triplets (i.e. hexamers). As a result, hexamer frequencies, usually in the form of codon position-dependent fifth-order Markov models (Box 5.2; Borodovsky and McIninch 1993), seem to offer the best

discriminative power to identify protein-coding regions in exons. Currently, these hexamer frequencies lie at the core of all modern eukaryotic gene predictors.

Exon Assembly

Once the exons are predicted (using a combination of hexamer frequencies and HMMs to identify key gene signals and exon/intron boundaries), they need to be assembled into some sort of multi-exon gene structure. The main difficulty in exon assembly lies in simple combinatorics: the number of possible exon assemblies grows exponentially with the number of predicted exons for any given gene. To address this problem, a number of dynamic programming techniques have been developed. Dynamic programming is an optimization technique that allows one to solve a complex problem by breaking it down into a collection of simpler subproblems. Each of those subproblems is solved just once, and their solutions are stored. The next time the same subproblem occurs, instead of recomputing its solution, one simply looks up the previously computed solution (Bellman 1957; see also Appendix 6.A for a detailed discussion). For the optimal exon assembly problem, dynamic programming has been shown to find the solution quite efficiently, without having to enumerate or consider each and every possible combination of exons (Gelfand and Roytberg 1993). Nearly all modern eukaryotic gene prediction tools now use some kind of dynamic programming method (called the Viterbi algorithm by Markov modelers, but also known as the Needleman–Wunsch algorithm by most people doing sequence alignment). By combining HMM-based exon signal identification with different HMM-derived scores for exons and then using dynamic programming to assemble the exons, it is possible to generate robust eukaryotic gene predictions. Some early examples of HMM-based gene prediction methods that use dynamic programming include GENIE (Kulp et al. 1996) and HMMgene (Krogh 1997). Perhaps the most popular example of an HMM-based eukaryotic gene predictor is GENSCAN (Burge and Karlin 1997), an *ab initio* gene predictor that has been widely used to annotate hundreds of eukaryotic genomes.

Given the popularity of GENSCAN, it is perhaps worthwhile explaining how this program works in a bit more detail and providing an example of how it can be used. For any given query sequence, GENSCAN determines the most likely gene structure given an underlying HMM. To model donor splice sites, GENSCAN introduced a method called *maximal dependence decomposition*. In this method, a series of weight matrices (instead of just one) are used to capture dependencies between positions in these splice sites. In addition, GENSCAN uses parameters that account for many higher order properties of genomic sequences (e.g. typical gene density, typical number of exons per gene, and the distribution of exon sizes for different types of exons). Separate sets of gene model parameters can be used to adjust for the differences in gene density and G + C composition seen across genomes. Models have also been developed for use with maize and *Arabidopsis* sequences. This leads to higher scores for exons exhibiting similarity to known proteins, but decreased scores for predicted exons having little to no similarity with known proteins.

A typical GENSCAN output is shown in Figure 5.5, using the human uroporphyrinogen decarboxylase (URO-D) gene (*U30787*) as the query. Each exon in the prediction is shown in a separate line. The columns, going from left to right, represent the gene and exon number (Gn . Ex), the type of prediction (Type, either the exon type or an identified polyA signal), the strand on which the prediction was made (+ or -), the beginning and endpoints for the prediction, the length of the predicted exon, its reading frame, several scoring columns, and a probability value (P). GENSCAN exons having a very high probability value ($p > 0.99$) are 97.7% accurate when the prediction matches a true, annotated exon. These high-probability predictions can be used in the rational design of polymerase chain reaction primers for complementary DNA (cDNA) amplification, or for other purposes where extremely high confidence is necessary. GENSCAN exons that have probabilities in the range of 0.50–0.99 are deemed to be correct most of the time. The best-case accuracies for p values higher than 0.90 is on the order of 88%. Any predictions having $p < 0.50$ should be deemed unreliable, and those data

GENSCAN Output

View gene model output: [PS](#) | [PDF](#)

GENSCAN 1.0 Date run: 26-May-118 Time: 18:37:33

Sequence /tmp/05_26_18-18:37:33.fasta : 4562 bp : 52.18% C+G : Isochore 3 (51 - 57 C+G%)

Parameter matrix: HumanIso.smat

Predicted genes/exons:

Gn.Ex	Type	S	.Begin	...End	.Len	Fr	Ph	I/Ac	Do/T	CodRg	P....	Tscr..
1.01	Intr	+	787	899	113	0	2	49	66	74	0.287	0.98
1.02	Intr	+	1796	1908	113	2	2	53	110	80	0.866	7.23
1.03	Intr	+	2024	2103	80	0	2	97	94	10	0.999	2.27
1.04	Intr	+	2180	2242	63	1	0	84	80	87	0.990	6.91
1.05	Intr	+	2482	2679	198	0	0	88	-9	263	0.895	16.67
1.06	Intr	+	2797	2958	162	0	0	107	109	97	0.965	14.39
1.07	Intr	+	3327	3464	138	2	0	52	77	126	0.812	9.07
1.08	Intr	+	3624	3724	101	2	2	87	119	113	0.996	13.71
1.09	Intr	+	3828	3894	67	0	1	63	77	46	0.998	0.40
1.10	Term	+	4227	4388	162	2	0	75	47	276	0.979	20.45
1.11	PlyA	+	4445	4450	6							1.05

Figure 5.5 Sample output from a GENSCAN analysis of the uroporphyrinogen decarboxylase gene. See the text for a more detailed description of the output.

are not given in the data table. The predicted amino acid sequence is given below the gene predictions. In the example shown here, GENSCAN correctly predicted nine of the 10 exons in URO-D. Only the initial exon was missed.

How Well Do Gene Predictors Work?

The accuracy of gene prediction programs is usually determined using controlled, well-defined datasets, where the actual gene structure has been determined experimentally. Accuracy can be computed at either the nucleotide, exon, or gene level, and each provides different insights into the accuracy of a predictive method. In the field of prokaryotic gene prediction, the results are almost always reported at the gene level and given in terms of a percentage – that is, the

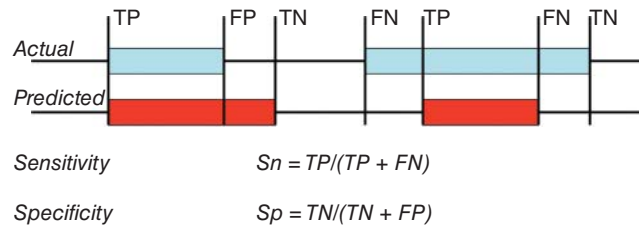


Figure 5.6 Schematic representation of measures of gene prediction accuracy at the nucleotide level. The actual gene structure is illustrated at the top with confirmed exons identified with light blue bars and confirmed introns in black lines. The predicted gene structure is illustrated at the bottom with predicted exons identified with red bars and predicted introns in black lines. The four possible outcomes of a prediction are shown: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The equations for sensitivity and specificity are also shown using appropriate combinations of TP, TN, FP, and FN.

number of correct gene predictions divided by the total number of known or validated genes in the test set. In some cases, the number or percentage of over-predicted genes (false positives) is also reported. In the field of eukaryotic gene prediction, performance reporting tends to be somewhat more convoluted. This is because the evaluation problem is more complex and the overall performance is often much worse. As a general rule, two basic measures are used: *sensitivity* (or S_n), defined as the proportion of coding nucleotides, exons, or genes that have been predicted correctly; and *specificity* (or S_p), defined as the proportion of coding and non-coding nucleotides, exons, or genes that have been predicted correctly (i.e. the overall fraction of the prediction that is correct). A more detailed explanation of sensitivity, specificity, and a number of other evaluation metrics used in gene (and protein structure) prediction is given in Box 5.4. Also introduced in this box are the concepts of true positives (TPs), true negatives (TNs), false positives (FPs), and false negatives (FNs).

An example of a eukaryotic gene prediction with the four possible outcomes is shown in Figure 5.6. This figure schematically illustrates the differences between a gene prediction and the known (or observed) gene structure. Neither sensitivity nor specificity alone provides a perfect measure of global accuracy, as high sensitivity can be achieved with little specificity and vice versa. An easier to understand measure that combines the sensitivity and specificity values is called the *Matthews correlation coefficient* (MCC or just CC), which is described more formally in Box 5.4. The MCC ranges from -1 to 1 , where a value of 1 corresponds to a perfect prediction; a value of -1 indicates that every coding region has been predicted as non-coding, and vice versa. Other accuracy measures are sometimes used as well; however, the above-mentioned ones have been most commonly employed in the large assessment projects on eukaryotic genome prediction such as the human ENCODE Genome Annotation Assessment Project (EGASP; Guigó and Reese 2005), the RNA-seq Genome Annotation Assessment Project (RGASP; Steijger et al. 2013) and the Nematode Genome Annotation Assessment Project (nGASP; Coghlan et al. 2008).

Box 5.4 Evaluating Binary Classifications or Predictions in Bioinformatics

Many predictions in bioinformatics involve essentially binary or binomial (i.e. true/false) classification problems. For instance, prokaryotic gene prediction can be framed as a binary classification problem where one tries to distinguish open reading frames (ORFs) from non-ORFs. Similarly, eukaryotic gene prediction can be posited as a binary classification problem of predicting exons and non-exons (introns) or genes and intergenic regions. Protein membrane helix prediction (discussed in Chapter 7) can be put in a similar binary classification frame, where one distinguishes between membrane helices and non-helices (or non-membrane regions). Binary classification problems can also be found in medicine, where one tries to predict or diagnose sick patients versus healthy patients, or quality control tasks in high-throughput manufacturing (pass versus fail).

The evaluation of binary classifiers or predictors normally follows a very standard practice with a common set of metrics and definitions. Unfortunately, this practice is not always followed when bioinformaticians evaluate their own predictors or predictions. This is why we have included this very important information box, one that is referred to frequently throughout this book.

As shown in the diagram below, a binary classifier or predictor can have four combinations of outcomes: true positives (TP or correct positive assignments), true negatives (TN or correct negative assignments), false positives (FP or incorrect positive assignments), and false negatives (FN or incorrect negative assignments). In statistics, the false positives are called type I errors and the false negatives are called type II errors (see Chapter 18).

		OBSERVATION	
		Observed state positive	Observed state negative
PREDICTION	Predicted state positive	True positive (TP)	False positive (FP)
	Predicted state negative	False negative (FN)	True negative (TN)

Once a binary classifier has been run on a set of data, it is possible to calculate specific numbers for each of these four outcomes using the above 2×2 contingency table. So a gene predictor that predicted 1000 genes in a genome that had only 900 genes may have 850 *TPs*, 200 *TNs*, 60 *FPs*, and 40 *FNs*. From this set of 4 outcomes it is possible to calculate 8 ratios. These ratios can be obtained by dividing each of the four numbers (*TP*, *TN*, *FP*, *FN*) by the sum of its row or column in the 2×2 contingency table. The most important ratios and their names or abbreviations are listed (along with their formulae) below. Also included are several other binary classifier evaluation metrics that are used by certain subdisciplines in bioinformatics or statistics.

Name	Formula
Sensitivity (<i>Sn</i>)	$\frac{TP}{TP + FN}$
Recall	
True positive rate (TPR)	
Specificity (<i>Sp</i>)	$\frac{TN}{TN + FP}$
True negative rate (TNR)	
Precision	$\frac{TP}{TP + FP}$
Positive predictive value (PPV)	
False positive rate (FPR)	$\frac{FP}{FP + TN}$
False discovery rate (FDR)	$\frac{FP}{FP + TP}$
Negative predictive value (NPV)	$\frac{TN}{TN + FN}$
Accuracy (ACC), Q_2	$\frac{TP + TN}{TP + FP + TN + FN}$
F1 score	$\frac{2TP}{2TP + FP + FN}$
F score	
F measure	
Matthews correlation coefficient (MCC)	$\frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$

(Continued)

Box 5.4 (Continued)

Sensitivity (S_n , recall, or TPR) measures the proportion of actual positives that are correctly identified as such, while specificity (S_p or TNR) measures the proportion of actual negatives that are correctly identified as such. Precision (PPV) is the proportion of positive results that are true positive results, while NPV is the proportion of negative results that are true negative results. FDR is the binary (not the multiple testing) measure of false positives divided by all positive predictions. Accuracy or ACC (for binary classification) is defined as the number of correct predictions made divided by the total number of predictions made. ACC is one of the best ways of assessing binary test or predictor accuracy. The F1 score is another measure of test accuracy and is defined as the harmonic average of precision (PPV) and recall (S_n). MCC is a popular measure of test or predictor accuracy. It is essentially a chi-squared statistic for a standard 2×2 contingency table. In effect, MCC is the correlation coefficient between the observed and predicted binary classifications.

Different fields of science have different preferences for different metrics owing to different traditions or different objectives. In medicine and most fields of biology (including bioinformatics), sensitivity and specificity are most often used to assess a binary classifier, while in machine learning and information retrieval, precision and recall are usually preferred. Likewise, different prediction tasks within bioinformatics tend to report performance with different measures. Gene predictors generally report S_n , S_p , and ACC, while protein structure predictors generally report ACC and MCC. The accuracy (ACC) score in protein secondary structure prediction has also been termed Q_n , where n is the number of secondary structure classes (usually $n = 3$). In gene prediction, the ACC score is given as Q_2 since only two classes are identified (either exon/intron or ORF/non-ORF). Each of the above ratios (except for MCC) can take on values from 0 to 1. For a perfect prediction, $S_n = 1$, $S_p = 1$, $PPV = 1$, $NPV = 1$, $ACC = 1$, $F1 = 1$, $MCC = 1$, $FPR = 0$, or $FDR = 0$, whereas a completely incorrect prediction would yield $S_n = 0$, $S_p = 0$, $PPV = 0$, $NPV = 0$, $ACC = 0$, $F1 = 0$, $MCC = -1$, $FPR = 1$, or $FDR = 1$.

The performance of any binary predictor has to be assessed based on the existing bias in the numbers within different classes (i.e. uneven class distribution). For example, an ACC of 0.95 may seem excellent, but if 95% of the dataset belongs to just one class, then the same ACC score could be easily achieved by simply predicting everything to be in that class. This is the situation with many mammalian genomes, which have large intergenic regions. Therefore predicting every nucleotide as being “intergenic” would easily create a nucleotide-based gene predictor that would be >95% accurate. Such a predictor would, of course, be completely useless.

Assessing the accuracy of gene prediction methods requires sets of reliably annotated genes verified by experimental or computational evidence derived from complementary sources of information. Experimental evidence can be provided by mass spectrometry-based proteomics or by structural biology methods such as nuclear magnetic resonance spectroscopy or X-ray crystallography (Chapter 12) that provide direct, visual confirmation of the protein sequences. Computational evidence can appear in the form of similarity of the derived protein sequence to the primary structures of proteins whose functions were verified experimentally. Extensive gene prediction assessments have been done for both prokaryotic and eukaryotic organisms.

Assessing Prokaryotic Gene Predictors

The evaluation of prokaryotic gene predictors has been ongoing for many years, with each publication that describes a new program (or a new version of an existing program) providing a detailed performance assessment (Larsen and Krogh 2003; Delcher et al. 2007; Hyatt et al.

2010; Borodovsky and Lomsadze 2011). One of the most recent and comprehensive assessments of prokaryotic gene predictors was conducted by Hyatt et al. in 2010. In this paper the authors compared five different programs: Prodigal 1.20 (Hyatt et al. 2010), GeneMarkHMM 2.6 (Borodovsky and Lomsadze 2011), GLIMMER 3.02 (Delcher et al. 2007), EasyGene 1.2 (Larsen and Krogh 2003), and MED 2.0 (Zhu et al. 2007) on two different tasks. The first task involved predicting experimentally verified genes with experimentally verified TISs from 10 different bacterial and archaeal genomes. In this case, only 2443 (out of a possible 35 000+) genes were considered experimentally verified. Hyatt et al. found that all five of the programs were able to achieve a 98–99.8% level of accuracy (at the gene level) for the 3' end of these verified genes and an 87–96.7% level of accuracy (at the gene level) for the complete genes (both the 5' and 3' ends being correctly predicted). The second task involved predicting GenBank (mostly manually) annotated genes from seven different bacterial genomes. In this case, a total of 23 648 genes were evaluated. All of the programs were able to achieve a 95–99% level of accuracy (at the gene level) for the 3' end of these genes. However, their performance for the full gene prediction task (both 5' and 3' being correctly predicted) was much more variable, with accuracy values ranging from 69% to 91% correct. The overall prediction average for all five programs over all genes in this second task was about 80%. It is also notable that all five programs generally over-predicted the number of genes annotated in GenBank by about 4–5%, with some programs (MED 2.0) over-predicting by as much as 40%.

Based on the data provided by Hyatt et al. (2010), the two best-performing prokaryotic gene prediction programs were Prodigal and GeneMark, while the other three programs were only marginally worse. Their results also show that the task of predicting the 3' ends of prokaryotic genes is essentially solved, while the challenge of predicting the 5' ends of prokaryotic genes needs more work. It is also evident that some prokaryotic genomes are harder to predict than others, with the full-gene prediction performance on the *E. coli* genome often hovering about 90% while the performance for less studied genomes (such as *Halobacterium salinarum*) often is around 70%. These results reflect the fact that ab initio gene predictors (both prokaryotic and eukaryotic) require very extensive training on a large number of high-quality gene models. Once trained, these tools can perform very impressively, especially in well-studied genomes for which ample training data are available. However, the level of training required to reach very high accuracy is often hard to achieve for newly assembled bacterial genomes.

Assessing Eukaryotic Gene Predictors

The assessment of eukaryotic gene predictors has been going on for more than 20 years. In the early days, most eukaryotic gene prediction evaluations were conducted on single genes whose exon/intron structure had been well characterized. This reflected the fact that very few (if any) eukaryotic genomes had been fully sequenced and only a small number of eukaryotic genes had their exon/intron structure fully determined. It also made the gene prediction tasks much simpler, as the coding (exon) density is much higher (25–50%) than what would be found over an entire genome (which is often <2%). This also led to overly optimistic performance ratings. More recently, the field has evolved to assessing gene prediction performance over entire genomes.

Burset and Guigó (1996) published one of the first systematic evaluations of eukaryotic gene predictors. Their study evaluated seven programs, using a set of 570 vertebrate single-gene sequences. The average CC at the nucleotide level for these programs ranged from 0.65 to 0.80. Later, Rogic et al. (2001) performed a similar analysis of seven gene prediction programs, using a set of 195 single-gene sequences from human and rodent species. The programs tested in the Rogic et al. study showed substantially higher accuracy than those reported on in the Burset and Guigó study, with the average CC at the nucleotide level ranging from 0.66 to 0.91. This increase in the upper part of the range illustrates the significant advances that occurred in the development of gene prediction methods over a relatively short period of time.

The early evaluations put forth by Burset and Guigó (1996), Rogic et al. (2001), and others all suffered from the same limitation: the gene finders were tested using controlled datasets comprising short genomic sequences encoding a single gene with simple gene structures. These datasets are obviously not representative of genomic sequences as a whole. Complete genome sequences contain long stretches with low coding density, stretches coding for multiple or incomplete genes (or both), and stretches having very complex or alternative gene structures. As a result, two large-scale studies were conducted to assess the performance of ab initio eukaryotic gene predictors on real-world mammalian genomic data. The first was based on an analysis of human chromosome 22 (Parra et al. 2003) and the second was based on an analysis of the human ENCODE regions (Guigó et al. 2006), covering about 1% of the human genome.

When human chromosome 22 was sequenced, it was subjected to very extensive manual analyses, experimental confirmation, and detailed annotation by many experts (Dunham et al. 1999). This was done to provide a useful gold standard (at the time) for assessing genome prediction and genome annotation tools. As a result, Parra et al. used the manually annotated data for chromosome 22 to assess the performance of GENSCAN (Burge and Karlin 1997), GenomeScan (Yeh et al. 2001), TBLASTX (Gish and States 1993), GeneID (Blanco et al. 2002), and SGP-2 (Parra et al. 2003) at the nucleotide, exon, and whole gene/transcript level. The results were quite disappointing. At the nucleotide level the programs had an average sensitivity/specificity ($(Sp + Sn)/2$) value ranging from 0.62 to 0.75 and a CC value ranging from 0.54 to 0.73. At the exon level, the programs had an average sensitivity/specificity value ranging from 0.54 to 0.62 and at the gene/transcript level the average sensitivity/specificity value ranged from 0.05 to 0.11. The latter values are the numbers of greatest interest as they reflect the true level of gene prediction performance. Interestingly, GENSCAN and GenomeScan performed somewhat worse than GeneID and SGP-2. Indeed, SGP-2 consistently performed better than all of the “pure” ab initio predictors as it also made use of comparative genomic data from mouse chromosome 22. The inclusion of experimental sequence data technically made SGP-2 an extrinsic gene finder rather than a pure ab initio gene predictor.

A similar level of very high-quality manual annotation was achieved in 2005–2006 during the first phase of the Encyclopedia of DNA Elements (ENCODE) project. The ENCODE project is a long-term, multi-phase project that started in 2003 with the goal of identifying all of the functional elements within the human genome sequence. During its pilot phase, a number of regions from the human genome (approximately 1%) were selected for detailed investigation. The availability of this “gold standard” dataset led to a second, much larger evaluation that looked at the predictive performance of pure ab initio predictors, as well as gene finders that used additional extrinsic data such as sequence homology and experimental sequencing data (Guigó et al. 2006). For the Guigó et al. study, four ab initio predictors were tested: AUGUSTUS (Hoff and Stanke 2013), GeneMark-A (Besemer and Borodovsky 2005), GeneMark-B (Besemer and Borodovsky 2005), and GeneZilla (Allen et al. 2006). Once again, the results were quite disappointing. At the nucleotide level, the programs had a CC value ranging from 0.53 to 0.76. At the exon level the programs had an average sensitivity/specificity value ranging from 0.40 to 0.57, and at the gene or transcript level the average sensitivity/specificity value ranged from 0.05 to 0.14. Overall, AUGUSTUS performed significantly better than the other ab initio programs but not at a level that would allow one to use it to automatically annotate a eukaryotic genome. However, the most important findings from this study were that significant improvements (up to two times better at the exon level and up to four times better at the gene level) could be made to the quality of eukaryotic gene annotations if comparative genomic data or other experimental/extrinsic evidence were employed in the prediction process.

It was because of these studies that a significant change in the gene prediction community occurred. In particular, the developers of gene predictors moved from reluctantly using experimental or extrinsic data to wholeheartedly embracing experimental data. In other words, gene *prediction* began to change to gene *finding* and genome prediction began to evolve toward genome *annotation*. In doing so, genome analysis became a more holistic, evidence-based process that combined ab initio gene prediction with extrinsic gene-finding methods. These

extrinsic gene-finding methods combined many other computational tools and other lines of evidence including gene expression data, proteomic data, sequence homology to other annotated genomes, and even literature-derived data.

Evidence Generation for Genome Annotation

Genomic evidence is any information that can be used to identify or inform the structure of a gene in an organism – be it a prokaryotic or a eukaryotic organism. Some of the most useful evidence comes from experimental work such as transcriptional data (mRNA or DNA data derived from RNA-seq experiments) or protein sequence data gathered about the organism of interest or a closely related organism. Other kinds of evidence can be collected through running various bioinformatic programs that identify genomic features such as sequence repeats, tRNA and rRNA genes, pseudogenes, transcription factor binding sites, retroviruses, prophages, and so on. In the following sections, we will briefly review some of the evidence-generating approaches used for both extrinsic gene finding and genome annotation.

Gene Annotation and Evidence Generation Using RNA-seq Data

RNA sequencing (RNA-seq) is a next-generation DNA sequencing (NGS) technique that involves converting RNA (mRNA, tRNA, and rRNA) transcripts into double-stranded cDNA fragments, then sequencing them using low-cost NGS sequencing methods (Wang et al. 2009). Over the last decade, RNA-seq has helped to revolutionize genome annotation methods for both eukaryotes and prokaryotes (Trapnell et al. 2009; Sallet et al. 2014). A typical RNA-seq experiment generates thousands of short DNA sequence reads corresponding to gene-coding regions (also known as coding sequence or CDS segments). These sequences can then be aligned to the reference genome sequence using gapped, short-read aligners to determine which genome regions were being transcribed. Some of the more popular gapped short-read aligners include TopHat2 (Kim et al. 2013), Stampy (Lunter and Goodson 2011), and GSNAP (Wu et al. 2016). These alignments can be further processed into putative transcripts using tools such as Cufflinks (Trapnell et al. 2012), StringTie (Pertea et al. 2015), or Trinity (Grabherr et al. 2011). In this way, RNA-seq provides experimental evidence (through DNA sequencing) regarding the location of gene-coding regions.

The improvement in gene-finding performance and gene annotation quality when RNA-seq data are used is quite substantial. In RGASP (Steijger et al. 2013), a diverse set of 14 genome annotation approaches were compared (including intrinsic/ab initio methods, extrinsic methods, and hybrid extrinsic/intrinsic methods). The gold standard for comparison was the reference human genome annotations from the GENCODE project, consisting of computationally, manually, and experimentally determined gene annotations (Harrow et al. 2012). It turned out that the best-performing programs for the task of identifying protein-coding genes were gene annotation tools that used RNA-seq data. Examples of gene annotation programs that incorporate RNA-seq data into their gene finders include AUGUSTUS (Hoff and Stanke 2013), mGENE (Schweikert et al. 2009), Trembly, and Transomics (Sperisen et al. 2004).

As noted earlier, when processing RNA-seq data for genome annotation, one can either splice-align the raw reads against the genome or, alternatively, transcript fragments can first be assembled *de novo* and then aligned to the genome via BLASTN. This “mapping-first” approach was shown to lead to more accurate annotations in the RGASP assessment and so it is highly recommended. Spliced alignment can be done with tools such as GSNAP (Wu et al. 2016), Stampy (Lunter and Goodson 2011), TopHat2 (Kim et al. 2013), or STAR (Dobin et al. 2013). Integrating coverage information from RNA-seq data into a gene annotation tool can typically be done by increasing the score of candidate exons that are covered by RNA-seq by a certain factor that depends on the local coverage of each covered exonic region. Rewarding

individual splice sites, supported by RNA-seq evidence, is relatively easy for HMMs. Some gene annotation tools integrate evidence for complete introns (i.e. splice site pairs) from RNA-seq data.

Newer technologies that produce longer RNA-seq reads (10 000+ bases) have greatly improved the ability to predict alternatively spliced transcripts in comparison with short reads (100–400 bases), which mostly help to find local alternative splice variants. Long reads are often near-complete transcripts and each spliced alignment gives the structure of a transcript, albeit only approximately owing to the relatively high sequencing error rate. Gene finders such as AUGUSTUS can integrate evidence from long-read alignments to further improve their performance.

While RNA-seq has greatly improved the performance of many eukaryotic gene finders, there is still a long way to go. According to the RGASP assessment (Steijger et al. 2013), the best-performing methods identified ~59% of protein-coding transcripts from the *Caenorhabditis elegans* genome (AUGUSTUS, mGene, and Transomics), 43% from the *Drosophila melanogaster* genome (AUGUSTUS), and just 21% from the *Homo sapiens* genome (Trembly). So, RNA-seq data have not (yet) been the key to “solving” the problem of accurate and automatic eukaryotic genome annotation. Important issues still remain, including the fact that a significant fraction of genes or splice forms may not be expressed in any RNA-seq sample, that transcribed sequences may not be protein coding and, if they are, the correct protein-coding ORFs remain to be identified, and that transcript assemblies and mapping of the transcripts to the genome are notoriously error prone. These errors typically are seen around exon boundaries, with the assemblies often extending into introns and, at times, missing whole exons. Several programs have been developed to help address these mapping problems, including Exonerate (Slater and Birney 2005) and GeneWise (Birney et al. 2004). Both programs are “splice-aware” tools that can be used to polish BLAST alignments. These polished alignments can then be used to improve the annotation of the exons, introns, splice sites, and 5' and 3' UTRs.

Gene Annotation and Evidence Generation Using Protein Sequence Databases

Just as RNA-seq data can be used as evidence for the existence of genes, so too can sequence homology be used to locate or identify new genes in newly sequenced organisms. In homology-based gene finding, the DNA sequence of the newly sequenced organism is translated into putative protein sequences and these putative sequences are then compared against databases of known proteins. Homologous matches at the protein level can then be used to annotate, identify, and locate the genes at the DNA level. A key advantage of homology-based gene finding over *ab initio* gene prediction is that homology-based methods provide not only the identification and location (as *ab initio* approaches do) but also the probable gene name and probable gene function as inferred by the sequence similarity of the newly identified gene to previously annotated proteins in the protein sequence databases.

Translated nucleotide searches such as BLASTX searches (Gish and States 1993) constitute one of the simplest homology-based gene prediction approaches. These searches are particularly useful when comparing ORFs in prokaryotic genomes. However, when dealing with the split nature of eukaryotic genes, BLASTX-like searches do not resolve exon splice boundaries particularly well. One useful approach is to use both the results of translated nucleotide searches along with those produced through the use of *ab initio* methods. Examples of this hybrid approach include programs such as GenomeScan (Yeh et al. 2001), GeneID (Blanco et al. 2002), and AUGUSTUS (Hoff and Stanke 2013). GenomeScan is an extension of GENSCAN that incorporates sequence similarity to known proteins using BLASTX.

A more sophisticated approach to eukaryotic gene prediction via sequence homology involves aligning the genomic query against a protein target that is presumed to be homologous to the protein encoded in the genomic sequence that is being annotated. In these alignments, often referred to as *spliced alignments*, large gaps corresponding to introns in the

query sequence are only allowed at “legal” splice junctions. Examples of programs using this approach include PROCUSTES (Gelfand et al. 1996), GeneWise (Birney and Durbin 1997), Exonerate (Slater and Birney 2005), BLAT (Kent 2002), and GenomeThreader (Gremme et al. 2005).

The spliced alignment approach does not exploit all the information typically available for homology-based gene prediction. In fact, for any given protein, a whole family of related proteins is often available. Such an ensemble of sequences carries more information than just a single protein. For instance, a well-constructed MSA shows which regions are well conserved and which ones are prone to insertions or deletions. Using an MSA, it is possible to calculate the probability that a certain amino acid occurs at a certain site. Using these data, it is possible to calculate PWMs or PSSMs and to create what is called an MSA profile. While the task for creating MSAs for prokaryotic genomes is relatively easy, the task of creating MSAs for eukaryotic genomes is particularly challenging owing to the presence of repeats, as well as large-scale genome rearrangements, duplications, and deletions.

So, rather than trying to find genes or exons with a set of individual protein sequences, one may use MSAs of aligned protein families to do the job. These MSAs can be found in orthology databases such as OrthoDB (Waterhouse et al. 2013). Several excellent software tools have been developed to search the gene structures of members of a protein family given an MSA profile representation of that family. These include GeneWise (Birney and Durbin 1997) and AUGUSTUS-PPX (Keller et al. 2011), where PPX stands for Protein Profile eXtension. AUGUSTUS-PPX has been shown to improve gene prediction accuracy over spliced alignment methods, especially when dealing with genes having large numbers of exons. However, the MSA approach is limited to the availability of homologous families and by the degree of sequence similarity. Therefore, MSA gene finding is best used in situations involving medium to high sequence similarity.

More recently, this MSA concept has been extended to cover situations with more remote sequence similarity through the development of BUSCO (Simão et al. 2015). BUSCO stands for Benchmarking Universal Single-Copy Orthologs. These single-copy orthologs correspond to a relatively small set of proteins that are highly conserved and that are found as single-copy genes across many different phyla in the tree of life. The BUSCO dataset currently includes 3023 genes for vertebrates, 2675 for arthropods, 843 for metazoans, 1438 for fungi, 429 for eukaryotes, and 40 universal marker genes for prokaryotes. Using HMMER (Eddy 2009), the BUSCO gene set can be rapidly searched against any given query genome. The presence or absence of these BUSCO genes in a given organism provides a good measure of the completeness of a genome assembly. It also provides a good measure of the completeness of a given genome annotation or a given genome prediction.

Gene Annotation and Evidence Generation using Comparative Gene Prediction

Another approach to homology-based gene prediction exploits the fact that there is a large and growing number of completely sequenced and well-annotated genomes now available. This has given rise to a technique called comparative gene prediction. The rationale behind comparative gene prediction is that functional regions (the protein-coding regions) tend to be more conserved than non-protein-coding regions. This observation provides the basis for identifying protein-coding regions in newly sequenced genomes. Comparative gene prediction methods exploit sequence homology but at a far more global scale than the protein sequence similarity methods described above. In comparative gene prediction, the “known” and “unknown” genomes are from different species, but the species are assumed to be so closely related that their entire genomes can be aligned. Because these genomes are so long (millions to billions of bases), the pairwise alignment or MSA is typically broken down into many local alignments of syntenic (homologous) regions.

Early methods for comparative gene finding typically used just two genomic sequences as input, such as DOUBLESCAN (Meyer and Durbin 2002), TWINSCAN (Korf et al. 2001), SLAM (Alexandersson et al. 2003), or SGP-2 (Parra et al. 2003). SLAM is an HMM-based method in which gene predictions and sequence alignments are performed simultaneously. TWINSCAN and DOUBLESCAN are extensions of GENSCAN, whereas SGP-2 is an extension of GeneID. Later on, comparative gene-finding methods were developed that could use more than two genomic sequences to predict genes in a new genome, but they only did so for a single target genome. These methods include programs such as N-SCAN (Gross and Brent 2006), CONTRAST (Gross et al. 2007), and Mugsy-Annotator (Angiuoli et al. 2011). More recently, a method called *clade annotation* has been developed and implemented in a version of AUGUSTUS known as “comparative AUGUSTUS” (König et al. 2016). Clade annotation allows the simultaneous alignment and annotation of multiple target genomes. For example, comparative AUGUSTUS can be used to simultaneously annotate the genomes of multiple (up to 20) different mouse strains.

Evidence Generation for Non-Protein-Coding, Non-Coding, or Foreign Genes

One of the best ways of determining the location of a protein-coding gene is to determine where it is not. In other words, knowing that a DNA segment cannot possibly code for a protein allows one to exclude it from the gene/protein-finding process. Prokaryotic genomes contain many genes that do not code for proteins. These include tRNA and rRNA genes, along with many foreign prophage genes that may or may not code for real phage proteins. Likewise, eukaryotic genomes are filled with repeat regions, pseudogenes, retrotransposons, and retroviral genes, along with an assortment of tRNA and rRNA genes. These non-protein-coding or non-coding elements can account for 20–30% of a given prokaryotic genome (Casjens 2003) and more than 90% of a eukaryotic genome (Li et al. 2004).

tRNA and rRNA Gene Finding

Both prokaryotes and eukaryotes have a significant portion of their genome occupied by tRNA and rRNA genes. Prokaryotes (including bacteria and archaea) typically contain 70–80 copies each of tRNA genes and between three and 45 copies of rRNA genes. tRNA molecules are L-shaped adaptor RNA molecules, typically 76–90 nucleotides in length, that are essential for the translational process (Figure 5.7). In principle, a total of 61 tRNA genes are needed to permit the translation of all 61 coding (sense) codons. However, because of a phenomenon known as base wobble, many organisms are able to have a single tRNA serving two or more codons. As a result, most prokaryotes have 35–40 unique tRNA genes, but one or two copies of each. rRNA molecules are the principal constituents (>60% by mass) of the ribosome, the translational engine of all cells. In prokaryotes the ribosome consists of two subunits, the small and the large subunit, which pair up to form the ribosome. The rRNAs present in prokaryotes are the 5S and 23S rRNAs in the large subunit and the 16S rRNA in the small subunit. Genes encoding these rRNAs are typically arranged into an operon (the *rrn* operon), with an internally transcribed spacer between the 16S and 23S rRNA genes. The number of *rrn* operons in prokaryotes ranges from one to 15 per genome.

tRNA and rRNA genes in eukaryotes share many similarities (in both structure and size) with those in prokaryotes. There are, however, some minor differences. For instance, eukaryotes typically have many more copies of tRNA genes than prokaryotes. There are 275 tRNA genes in *Saccharomyces cerevisiae*, 620 copies of tRNA genes in *C. elegans*, and 497 copies of tRNA genes in humans. All eukaryotes have 22 mitochondrial tRNA genes. Like prokaryotes, eukaryotic rRNA genes are also divided according to their location in the large or small subunit in the ribosome. However, instead of just two large subunit rRNAs, there are three rRNAs in the eukaryotic large subunit: 5S, 5.8S, and 28S. Just as with prokaryotes, eukaryotes have one rRNA gene (18S rRNA) for their small ribosomal subunit, but they also encode rRNA genes

Figure 5.7 The typical L-shaped structure of a tRNA molecule. This depicts the three-dimensional structure of the yeast phenylalanine tRNA molecule at 1.93 Å resolution (Protein Data Bank (PDB) accession code: 1EHZ).



for their mitochondrial ribosomes (12S and 16S rRNA genes). Unlike prokaryotes, eukaryotes generally have many copies of the rRNA genes organized into tandem repeats. In humans, approximately 300–400 rRNA repeats are present in five clusters, located on five separate chromosomes. Unlike prokaryotic tRNA genes, some tRNA genes in eukaryotes are interrupted with introns.

The structure of tRNAs is highly conserved across all major kingdoms of life and there are a large number of tRNA sequences that are known for both prokaryotes and eukaryotes. As a result, most methods for identifying tRNA genes take advantage of common sequence motifs (recognizable via HMMs) and employ some kind of sequence homology or database comparison to identify tRNA genes. The best performing and most popular methods are RNAMotif (Macke et al. 2001), tRNAfinder (Kinouchi and Kuoakawa 2006), and tRNAscan-SE (Lowe and Eddy 1997). These programs are able to identify tRNA genes in both prokaryotes and eukaryotes with very high accuracy (>95%). In addition to these programs, there are several dedicated databases of tRNA sequences to assist with comparative tRNA identification approaches; these include tRNAdb and tRNADB-CE (Jühling et al. 2009; Abe et al. 2014). Currently the identification of tRNA genes is considered to be a “solved” problem.

Just like tRNA genes, rRNA genes also exhibit a very high level of sequence conservation, and there are a number of rRNA motifs that can be described by HMMs. These HMMs have been integrated into the program (and web server) called RNAmmer (Lagesen et al. 2007). RNAmmer is able to identify all rRNAs from prokaryotes and eukaryotes, with the exception of 5.8S rRNA. Owing to the complexity, size, and relatively poor annotation of rRNA genes, the performance for rRNA prediction is not yet at the same level as tRNA prediction. In addition to the RNAmmer predictor, there is also an RNA database called Rfam (Kalvari et al. 2018) that contains >2600 RNA families (including rRNA and tRNA sequence families). Each sequence family in Rfam is represented by an MSA, a consensus secondary structure, and a covariance model. Rfam can be used for rRNA (and other RNA gene) identification via sequence comparisons or MSAs. Regardless of their current shortcomings, the use of tRNA and rRNA gene identification tools invariably improves the accuracy of any protein-coding gene-finding or gene prediction effort. It also enhances the quality of the overall genome annotation.

Prophage Finding in Prokaryotes

Prokaryotes are subject to constant attack by bacterial viruses called bacteriophages, which kill or disable susceptible bacteria. Bacteriophages are the most abundant biological entities on the planet and they play a major role in the bacterial ecosystem and in driving microbial genetic variation or genetic diversity. This genetic diversity is brought on through a particularly

unique part of the bacteriophage life cycle called lysogeny. Lysogeny involves the integration of the phage genome (often consisting of 10–20 genes) into the host bacterial chromosome at well-defined insertion points. The genetically integrated phages are called prophages. In some cases, prophages can become permanently embedded into the bacterial genome, becoming cryptic prophages (Little 2005). These cryptic prophages often serve as genetic “fodder” for future evolutionary changes of the host microbe (Bobay et al. 2014). Furthermore, prophages and cryptic prophages tend to introduce pathogenic elements or pathogenic islands that exhibit a very different base composition than the host genome. Prophages and cryptic prophages can account for up to 20% of the genetic material in some bacterial genomes (Casjens 2003), with some prophage genes coding for expressed proteins and others not.


Given their high abundance, the identification of these phage-specific genetic elements can be quite important, especially when it comes to annotating bacterial genomes. Prophage and cryptic prophage sequences exhibit certain sequence features (such as the presence of integrases and transposases, attachment sites, and an altered base composition) that can be used to distinguish them from “normal” bacterial genes. When combined with HMMs to improve the sequence feature recognition tasks, it is possible to identify prophage and cryptic prophage sequences with relatively good accuracy. The accuracy can be improved further if comparative genome analyses to databases of known phage sequences are performed. Several bacterial prophage-finding programs have been developed and deployed over the past decade, including Phage_Finder (Fouts 2006) and Prophage Finder (Bose and Barber 2006). More recently, phage finding has moved from stand-alone programs to web servers. In particular, two new web servers have been released that provide somewhat greater speed and improved accuracy for prophage finding over existing tools. These are known as PHAST (Zhou et al. 2011) and PHASTER (Arndt et al. 2016). Both web servers are between 85% and 95% accurate (depending on the test being conducted) and both provide rich graphical output, as well as detailed annotations of the prophage sequences and the surrounding bacterial genomic sequences (Figure 5.8). Regardless of the method chosen, the annotation of prophage and cryptic prophage genes certainly enhances the quality of a prokaryotic genome annotation and it usually improves the accuracy of any prokaryotic gene predictions.

Repetitive Sequence Finding/Masking in Eukaryotes

Unlike prokaryotes, eukaryotic genomes contain considerable quantities of repetitive DNA. These repetitive sequences include retrotransposons and DNA transposons, both of which are referred to as dispersed repeats, as well as highly repetitive sequences, typically called tandem repeats. The most abundant repeat sequences in eukaryotes are retrotransposons. Retrotransposons are genetic elements that can amplify themselves using a “copy-and-paste” mechanism similar to that used by retroviruses. To replicate and amplify, they are transcribed into RNA, then converted back into identical DNA sequences using reverse transcription and then inserted into the genome at specific target sites. In contrast to retrotransposons, DNA transposons do their copying and pasting without an RNA intermediate, instead using the protein called transposase. Approximately 52% of the human genome is made up of retrotransposons, while DNA transposons account for another 3% (Lander et al. 2001; Wheeler et al. 2013). In plants, retrotransposons are much more abundant, accounting for between 60% and 90% of the DNA in any given plant genome (Li et al. 2004).

Within the retrotransposon family are two subfamilies: long terminal repeat retrotransposons (LTR retrotransposons) and non-LTR retrotransposons. LTR retrotransposons are retrovirus-like sequences that contain LTRs that range from ~100 bp to over 5 kb in length. In fact, a retrovirus can be transformed into an LTR retrotransposon simply through inactivation or deletion of certain genes (such as the envelope protein) that enable cell-to-cell viral transmission. Most LTR retrotransposons are non-functional endogenous retroviruses that are also called proviruses. In this regard, eukaryotic LTR retrotransposons can be thought of as the equivalent of prokaryotic prophages or cryptic prophages. Human endogenous

PHASTER New Search Genomes Help About My Searches



PHASTER

PHASTER (PHAge Search Tool Enhanced Release) is a significant upgrade to the popular PHAST web server for the rapid identification and annotation of prophage sequences within bacterial genomes and plasmids. While the steps in the phage identification pipeline in PHASTER remain largely the same as in the original PHAST, numerous software improvements and significant hardware enhancements have now made PHASTER faster, more efficient, more visually appealing and much more user friendly. In particular, PHASTER is now 4.3X faster than PHAST when analyzing a typical bacterial genome. More specifically, software optimizations have made the backend of PHASTER 2.7X faster than PHAST. Likewise, the addition of more than 120 CPUs to the PHASTER compute cluster have greatly reduced processing times. PHASTER can now process a typical bacterial genome in 3 minutes from the raw sequence alone, or in 1.5 minutes when given a pre-annotated GenBank file. A number of other optimizations have been implemented, including automated algorithms to reduce the size and redundancy of PHASTER's databases, improvements in handling multiple (metagenomic) queries and high user traffic, and the ability to perform automated look-ups against >14,000 previously PHAST/PHASTER annotated bacterial genomes (which can lead to complete phage annotations in seconds as opposed to minutes). PHASTER's web interface has also been entirely rewritten. A new graphical genome browser has been added, gene/genome visualization tools have been improved, and the graphical interface is now more modern, robust, and user-friendly.

Please cite the following:

Arndt, D., Grant, J., Marcu, A., Sajed, T., Pon, A., Liang, Y., Wishart, D.S. (2016) PHASTER: a better, faster version of the PHAST phage search tool. *Nucleic Acids Res.*, 2016 May 3.

Zhou, Y., Liang, Y., Lynch, K.

Output File Formats

Summary File :

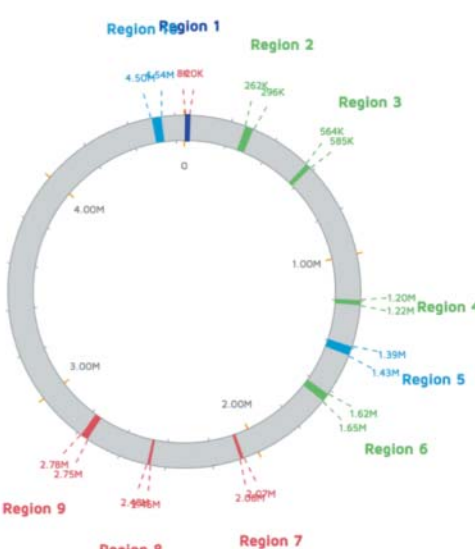
gi|49175990|ref|NC_000913.2| Escherichia coli str. K-12 substr. MG1655, complete genome. 4639675, gc%: 50.79%

Download summary as .txt file: [summary.txt](#)

Total: 10 prophage regions have been identified, of which 4 regions are intact, 4 regions are incomplete, and 2 regions are questionable.

Region	Region Length	Completeness	Score	# Total Proteins	Region Position	Most Common Phage	GC %	Details
1	12Kb	incomplete	30	13	8238-20314	PHAGE_Cafete_BV_PW1_NC_014637(2)	50.51%	Show
2	34.3Kb	intact	120	42	262122-296430	PHAGE_Enterococcus_P1_NC_005856(2)	53.45%	Show
3	21.3Kb	intact	130	35	563978-585280	PHAGE_Enterococcus_lambda_NC_001416(9)	43.27%	Show
4	19.8Kb	intact	110	26	1196090-1215894	PHAGE_Shigella_SfV_NC_022749(12)	43.66%	Show
5	39.1Kb	questionable	90	33	1393976-1433075	PHAGE_Escherichia_HK639_NC_016158(4)	47.99%	Show
6	34.3Kb	intact	130	43	1617581-1651939	PHAGE_Enterococcus_mEp460_NC_019716(5)	45.59%	Show
7	13.5Kb	incomplete	50	16	2088978-2089371	PHAGE_Stx2_converting_1717_NC_011357(2)	52.15%	Show
8	11.2Kb	incomplete	50	18	2454376-2475651	PHAGE_Shigella_SfV_NC_022749(13)	44.39%	Show
9	31.5Kb	incomplete	30	32	2749817-2781326	PHAGE_Enterococcus_P4_NC_001609(1)	47.08%	Show
10	596.1Kb	intact	100	343	596100-1192200	PHAGE_Enterococcus_005344(2)	48.86%	Show

Escherichia coli str. K-12 substr. MG1655



Prophage Region 1

Start: 8238
End: 20314
CDS: 13
Predicted Type: incomplete
GC%: 50.51

■ Intact (score > 90)
■ Questionable (score 70-90)
■ Incomplete (score < 70)

Viewer Options

Hide Region Labels

Show Label Lines

Hide Markers

Save Image

Length: 4639675 bps
Phages: 10

Figure 5.8 A screenshot montage of the PHASTER web server showing the website homepage along with examples of some of PHASTER's output.

retroviral sequences, all of which appear to be defective or non-replicative, account for about 8% of the human genome (Taruscio and Mantovani 2004).

Non-LTR retrotransposons consist of two subtypes: long interspersed nuclear elements (LINEs) and short interspersed nuclear elements (SINEs). LINEs are typically 7000 bp long and encode several genes to cover all the functions needed for retrotransposition. These include reverse transcriptase and endonuclease genes, as well as several genes needed to form a ribonucleoprotein particle. More than 850 000 copies of LINEs exist in the human genome, covering 21% of all human DNA (Cordaux and Batzer 2009). However, more than 99% of LINEs are genetically “dead,” having lost their retrotransposition functions. In contrast to LINEs, SINEs are much smaller, typically consisting of DNA stretches spanning just 80–500 bp. SINEs are very abundant (in the millions of copies), accounting for about 10% of the DNA in the human genome. The most common SINEs in humans are the Alu repeats (Häsler and Strub 2006). Alu repeats are about 300 bp long. They are highly conserved in primates and are subject to frequent DNA methylation events.

In addition to transposable elements (or dispersed repeats), eukaryotes also contain large numbers of tandem repeats, including minisatellite DNA, microsatellite DNA (also known as short tandem repeats [STRs] or simple sequence repeats [SSRs]), and telomere repeats. Minisatellite DNA consists of repeats of 10–60 bp in length that stretch for about 2 kb and are scattered throughout the genome. Microsatellite DNA consists of repeats of 1–6 bp, extending for hundreds of kilobases, particularly around the centromeres. Telomere repeats consist of a highly conserved 6 bp sequence (TTAGGG) that is repeated 250–1000 times and found exclusively at the ends of eukaryotic chromosomes. Mini- and microsatellite DNA account for about 5% of the DNA in the human genome (Subramanian et al. 2003).

The fact that eukaryotes have so many repeat sequences, combined with the fact that these repeats account for such a large portion of their genomes (often >50%), has led to concerted efforts by genome annotators to identify, remove, or mask these sequences. This is because repeat sequences can seriously hinder gene identification and genome annotation activities. For instance, retrotransposons and DNA transposons can be easily mistaken as exons by *ab initio* gene predictors. Likewise, STRs can lead to spurious alignments when comparative genomics approaches are used in gene finding. STRs (which are also called low-complexity regions) can often be dealt with through two techniques: soft or hard masking. Soft masking is done by changing the case of the letters in the sequence file from upper case to lower case, while hard masking changes the offending sequence to Ns, thereby removing them completely from consideration. Soft masking prevents the masked region from seeding alignments but preserves sequence identity so that off-target alignments are minimized. Soft masking is routinely done by the programs SEG and DUST (Wootton and Federhen 1993), which are found in most versions of the BLAST sequence alignment suite.

While tandem repeats are relatively easy to deal with, repeat transposable elements (such as retrotransposons) are much harder to handle. This is because these sequences are far larger and far more complex. The Repbase (Jurka et al. 2005) database contains a comprehensive collection of repeat and transposable elements from a wide range of species. This resource is frequently used to identify repeat elements via comparative sequence analysis. However, if the transposon sequences are highly divergent from those found in RepBase, then other methods or other databases may need to be used. Dfam (Wheeler et al. 2013) is an example of a more advanced repetitive element database. In Dfam, the original Repbase sequences have been converted to HMMs. The use of these HMMs has allowed many more transposable elements to be identified (up to 54.5% vs. 44% in humans) with much improved accuracy (Wheeler et al. 2013). In addition to Dfam (which is available as both a server and a downloadable resource), there are several stand-alone programs and web servers that have been developed to specifically identify retrotransposons, including RECON (Bao and Eddy 2002), RepeatScout (Price et al. 2005), RetroPred (Naik et al. 2008), LTR_FINDER (Xu and Wang 2007), LTRharvest (Ellinghaus et al. 2008), and MITE-Hunter (Han and Wessler 2010). These programs identify and label either LTR or non-LTR retrotransposons. While this information may be useful to some, many

genome annotators are simply interested in removing retrotransposons from consideration. In this regard, RepeatMasker (Tarailo-Graovac and Chen 2009) has become the tool of choice as it simply hard masks (i.e. removes) all detectable retrotransposon sequences from the genome of interest.

As a general rule, hard masking of transposable elements is often the first step performed in a eukaryotic genome annotation. Hard masking with tools such as Dfam or programs such as RepeatMasker not only removes “uninteresting” genetic data, it also accelerates the gene identification process and improves annotation accuracy. Because coding exons tend not to overlap or to contain repetitive elements, ab initio gene prediction programs tend to predict fewer false-positive exons when using hard-masked sequences. For instance, when chromosome 22 was analyzed using different ab initio gene predictors, it was found that a significant reduction in false-positive gene predictions occurred (Parra et al. 2003). In particular, GENSCAN initially predicted 1128 protein-coding genes without using sequence masking, but when sequence masking was used the number of predicted genes dropped to 789. When GeneID was used, the number fell from 1119 to 730. The actual number of protein-coding genes in chromosome 22, according to the latest GENCODE annotation, is 489.

Finding and Removing Pseudogenes in Eukaryotes

A particular challenge with eukaryotic genome annotation is differentiating between predictions identifying “real” genes from those that correspond to non-functional pseudogenes. Database searches may not help to provide any clearer picture, as many pseudogenes are similar to functional, paralogous genes. The absence of an RNA transcript from an RNA-seq experiment cannot be used as a criterion either, because RNA transcripts do not always exist for actual genes because of variations in tissue expression or developmental stages. In general, intronless gene predictions for which multi-exon paralogous genes exist in the same genome are suspicious, as they may indicate sequences that have arisen through retrotransposition. Multi-exon predictions, however, can also correspond to pseudogenes arising through a recent gene duplication event. If homologs in another organism exist, one solution is to compute the synonymous versus non-synonymous substitution rate (K_a/K_s ; Fay and Wu 2003). K_a/K_s values approaching 1 are indicative of neutral evolution, suggesting a pseudogene. Support for multi-exon gene predictions can come from assessing the conservation of the overall gene structure in close homologs. For instance, the prediction or identification of homologous genes in two modestly related organisms (e.g. mouse and human) most likely indicates that the gene is real and is not a pseudogene (Guigó et al. 2003).

Genome Annotation Pipelines

In the early days of genome annotation, when it would often take years just to sequence a single organism, teams of researchers and bioinformaticians would gather and work together for many months, or even years, to assemble the genome, perform the initial ab initio gene predictions, manually collate the experimental or literature-derived evidence, conduct comparative sequence analysis, and then synthesize the data into a consensus genome annotation. This was routinely done for both bacterial and eukaryotic genomes (Lander et al. 2001; Winsor et al. 2005; Riley et al. 2006). Indeed, it is still being done for the GENCODE project, which has been preparing and updating the reference human genome annotation since 2003 (Harrow et al. 2012). However, these efforts have required (and continue to require) enormous resources and time. With the appearance of very high-throughput NGSs and the ability to routinely sequence an entire genome in a few days, these manual approaches to genome annotation have become unsustainable. Now, most genome annotations are done through automated pipelines that help users to synthesize multiple pieces of evidence and data to generate a consensus genome annotation.

The choice of a pipeline tool depends on the type of organism (eukaryote vs. prokaryote), the computational resources available, the available evidence (RNA-seq or no RNA-seq data), and the similarity of the organism to previously annotated organisms. For instance, if one is annotating a genome with a closely related, previously annotated species, a simple comparative analysis or sequence projection should be sufficient. If the organism of interest has no closely related annotated species, a pipeline that uses RNA-seq or experimentally acquired protein sequence data will generate more accurate annotations. The most advanced genome annotation pipelines require many programs and perform complex analyses that need supercomputers such as large multi-core machines or massive computing clusters (maintained locally or available via the Cloud). For example, to annotate the loblolly pine genome (which contains 22 billion bases – seven times more than the human genome) required 8640 central processing units (CPUs) running for 14.6 hours (Wegrzyn et al. 2014). In the following sections we will briefly describe some commonly used annotation pipelines for prokaryotes and eukaryotes.

Prokaryotic Genome Annotation Pipelines

Annotation pipelines for prokaryotes typically do not require the same computational resources as for eukaryotes. Indeed, most bacterial genomes can be annotated in less than 30 minutes, whether on a web server or on a desktop computer. However, the recent shift toward metagenomics or community bacterial genomics is beginning to lead to significantly greater computational demands that will be discussed in more detail in Chapter 16. Some of the more popular publicly available prokaryotic genome annotation pipelines include Prokka (Seemann 2014), Rapid Annotation using Subsystem Technology (RAST; Overbeek et al. 2014), and the Bacterial Annotation System (BASys; Van Domselaar et al. 2005). Prokka is an open-source Perl program that runs with a command-line interface (on UNIX). Prokka can be used to annotate pre-assembled bacterial, archaeal, and viral sequences. With Prokka, a typical 4 million base pair bacterial genome can be fully annotated in less than 10 minutes on a quad-core computer. Prokka is also capable of producing standards-compliant output files for further analysis or viewing. Prokka's appeal lies in its speed and ability to perform "private" annotations on local computers. In contrast to Prokka, RAST and BASys are genome annotation web servers. Web servers are generally easier to use but they do not offer the privacy of a locally installed program. RAST is a registration-based web server that accepts standard, pre-assembled DNA sequence files and then identifies protein-encoding, rRNA, and tRNA genes, assigns functions to the genes, and finally uses this information to reconstruct a metabolic network for the organism. In contrast to RAST, BASys is an open access web server. BASys accepts pre-assembled FASTA-formatted DNA or protein files from bacteria, archaea, and viruses and performs many of the same annotation functions as RAST. However, BASys provides a much greater depth of annotation (covering more than 50 calculable properties) and produces colorful, easily viewed genome maps (Figure 5.9) using a program called CGView (Stothard and Wishart 2005).

Eukaryotic Genome Annotation Pipelines

Given the complexity of eukaryotic genomes, their corresponding annotation pipelines must do somewhat more than those used for prokaryotic genomes. In particular, eukaryotic genome annotation pipelines must combine not only the *ab initio* gene predictions (or multiple gene predictions from multiple sources) but also many other pieces of evidence, including experimental data. As a result, almost all modern eukaryotic genome annotation pipelines use a technique called "evidence clustering" to identify gene regions and then use the aligned RNA (from RNA-seq) and protein evidence to improve the accuracy of the gene predictors. Some pipelines go even further and make use of a "combiner" algorithm to select the combination of exons that are best supported by the evidence. Two combiner programs in particular are very good at this: JIGSAW (Allen and Salzberg 2005) and EVidenceModeler, or EVM (Haas

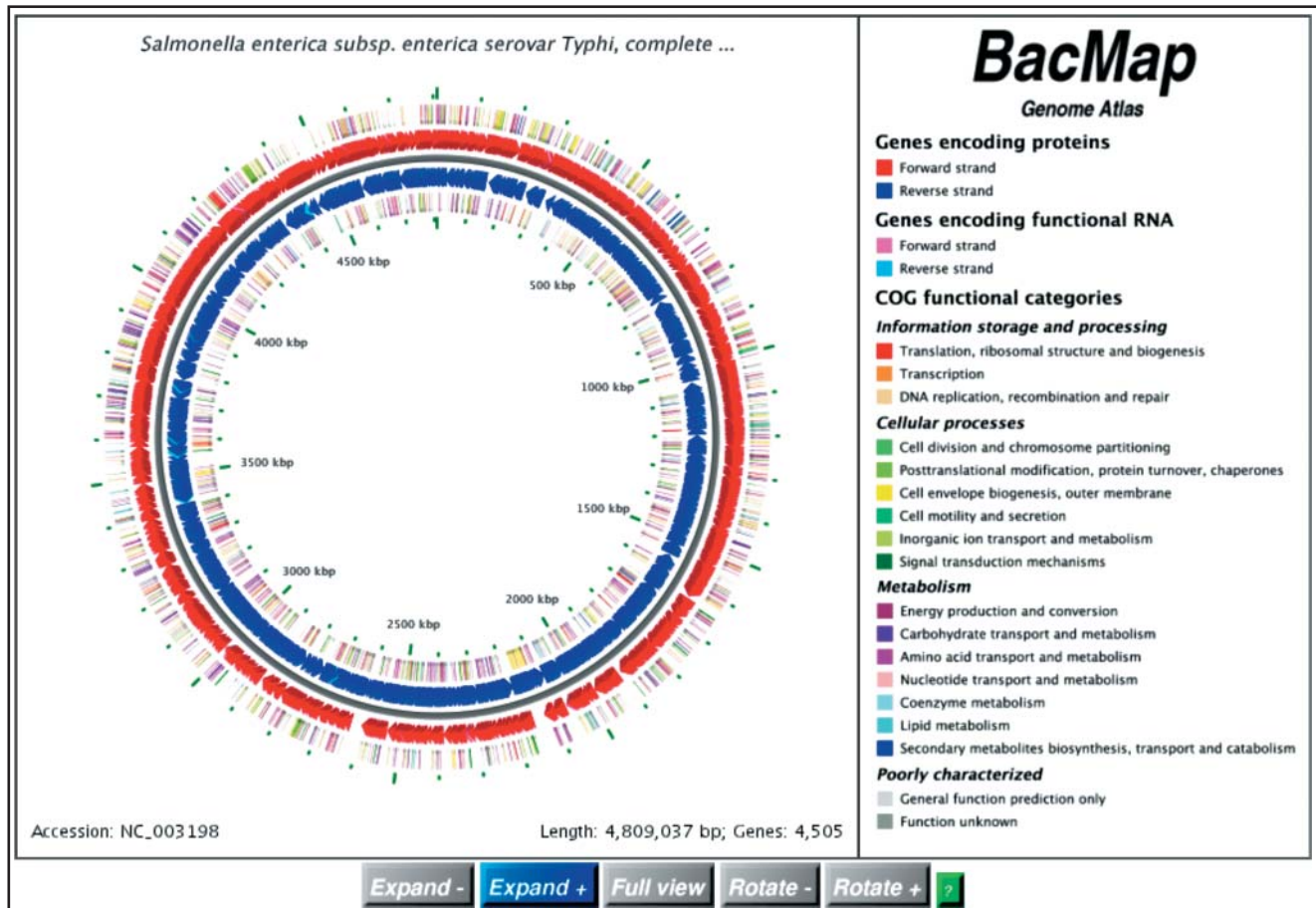


Figure 5.9 A screenshot of a BASys bacterial genome annotation output for the bacterium *Salmonella enterica*. The BASys image can be interactively zoomed-in to reveal rich annotations for all of the genes in the genome.

et al. 2008). These programs assess different types of evidence based on known error profiles and various kinds of user input and then choose the best combination of exons to minimize the error. In particular, EVM combines aligned protein and RNA transcript evidence with ab initio gene predictions into weighted consensus gene models, while JIGSAW uses non-linear models or weighted linear combiners to choose a single best consensus gene model.

Among the most widely used eukaryotic genome annotation pipelines, all of which use some kind of combiner algorithm, are MAKER2 (Holt and Yandell 2010), Ensembl (Fernández-Suárez and Schuster 2010), the National Center for Biotechnology Information (NCBI) Eukaryotic Annotation Pipeline (Thibaud-Nissen et al. 2016), PASA (Haas et al. 2008), and BRAKER1 (Hoff et al. 2016). The MAKER2 annotation pipeline is a highly parallelizable, stand-alone program that aligns and polishes protein sequence and transcriptome (RNA-seq) data with BLAST; it also provides evidence-based hints to various gene predictors and it creates an evidence trail with various quality metrics for each annotation. Some of MAKER2's quality metrics include the number of splice sites confirmed by RNA-seq evidence, the number of exons confirmed by RNA-seq data, and the lengths of 5' and 3' UTRs. MAKER2 also uses a quality metric called the Annotation Edit Distance, or AED (Eilbeck et al. 2009). The AED value ranges between 0 and 1, with higher quality annotations being associated with lower AEDs. MAKER2 uses these AED values to choose the best gene predictions from which to build its final annotation. Like the MAKER2 pipeline, the Ensembl genome annotation pipeline builds its gene models from aligned and polished protein sequence- and RNA-seq-derived transcriptome data. To complete the annotation process, Ensembl merges

identical transcripts, and a non-redundant set of transcripts is reported for each gene. Both MAKER2 and Ensembl supply hints to indicate intron/exon boundaries using protein and RNA-seq alignments to their internal gene predictors. This helps to generate gene models that better represent the aligned evidence. This approach also helps improve gene prediction accuracy for poorly (or insufficiently) trained gene finders. Like the Ensembl and MAKER2 pipelines, the NCBI Annotation Pipeline aligns and polishes protein and transcriptome data. It also generates gene predictions using the Gnomon gene-finding program (Souvorov et al. 2010). The NCBI system typically assigns higher weights to manually curated evidence over computationally derived models or computationally generated evidence. The PASA genome annotation pipeline is one of the oldest annotation pipelines and was one of the first to use a combiner or evidence-clustering algorithm (EVM). PASA aligns RNA transcripts to the reference genome using BLAT (Kent 2002) or GMAP (Wu et al. 2016). PASA is capable of generating annotations based on RNA transcriptome data, on pre-existing gene models, or on *ab initio* gene predictions. PASA, along with the MAKER2 and Ensembl annotation pipelines, are able to add UTRs to their genome annotations via RNA-seq data to further increase their accuracy. One of the latest additions to publicly available eukaryotic genome annotation pipelines is the BRAKER suite of programs (Hoff et al. 2016). BRAKER1 (and most recently BRAKER2) combines the strengths of GeneMark-ET with AUGUSTUS – both of which use RNA-seq data to improve their gene annotation accuracy. In the BRAKER pipeline, GeneMark-ET is used first to train and generate initial gene structures, then AUGUSTUS makes use of the initially predicted genes for further training and integrates RNA-seq data into the final gene predictions. BRAKER1 has been shown to be 10–20% more accurate than MAKER2 in terms of gene and exon sensitivity/specificity.

Even with an exon accuracy of >90% (rarely achieved by even the best eukaryotic genome annotation pipelines), most genes in a genome will have at least one incorrectly annotated exon. Incorrectly identified genes or mistaken gene annotations can have very serious consequences for experimentalists who are designing experiments to study gene functions. Indeed, many failed molecular biology or gene-cloning experiments can be traced back to incorrect gene annotations. Furthermore, incorrect annotations can propagate, leading to a cascade of errors that affect many other scientists. This happens when an incorrect annotation is innocently passed on to another genome project and then used as evidence in still more genome annotation efforts which eventually end up in public databases. To help prevent these errors or to reduce the magnitude of these mistakes, most annotation pipelines include some kind of quality metrics which are attached to each and every gene annotation. Most of these metrics are based on a score that measures the agreement of a given gene annotation to an aligned RNA/protein sequence or on the basis of the homology and synteny of the gene to closely related species. Some pipelines use a simple star rating (ranging from zero to five). Zero stars corresponds to an annotation where none of the exons is supported by aligned evidence, while a five-star rating corresponds to a situation where every exon is supported and every splice site is confirmed by a single full-length cDNA. Other pipelines use more sophisticated metrics, such as the AED score (mentioned above). Protein family domains can also be good indicators of annotation quality and annotation completeness. Certainly, any annotation that contains an identifiable protein domain is more likely to encode a functional protein than one that does not. Domain matching has been used to rescue a number of gene annotations that would have otherwise received a “failing” quality score owing to a poor sequence alignment. Both Ensembl and MAKER2 report the fraction of annotations containing a protein family domain as a quality measure. Interestingly, this fraction (0.69) appears to be quite constant across genomes; the closer a given genome is to this fraction, the more confidence one has in its quality. In addition to the domain-matching fraction, the presence or absence of BUSCO genes can also be used to provide a measure of the completeness of a genome annotation (Simão et al. 2015). Another excellent route to ensure good quality annotations is through manual inspection with genome visualization and editing software. This is discussed in more detail below.

Visualization and Quality Control

While automated or semi-automated pipelines for genome annotation have become the norm, there is still a need for a human factor in annotating genomes and assessing their quality. Having a knowledgeable biologist or some kind of “domain expert” carefully look through a genome annotation is essential to ensure that the annotations make sense. This manual review process also allows one to catch and correct suspicious annotations or fill in missing annotations. However, to perform these manual reviews or curatorial tasks, it is necessary to visualize and interactively edit the annotations. Certainly two of the best known genome browsers are the University of California Santa Cruz Genome Browser (Casper et al. 2018) and Ensembl’s Genome Browser (Fernández-Suárez and Schuster 2010), both of which have been thoroughly reviewed in Chapter 4. While these tools are excellent for visualizing genome annotations, there are also a number of other tools that support both visualizing and editing genome annotations, including Web Apollo (Lee et al. 2013), GenomeView (Abeel et al. 2012), and Artemis (Carver et al. 2012).

Web Apollo is both a visualization tool and a genome editor. More specifically, it is a web-based plug-in for JBrowse (Westesson et al. 2013) that provides an editable, user-created annotation track. All edits in Web Apollo are visible in real time to all members of the annotation team. This feature is particularly helpful when undertaking a community annotation project or when many investigators are involved in a particular genome analysis. GenomeView is an open-source, stand-alone genome viewer and editor that allows users to dynamically browse large volumes of aligned short-read data. It supports dynamic navigation and semantic zooming, from the whole genome level to the single nucleotide level. GenomeView is particularly noted for its ability to visualize whole genome alignments of dozens of genomes relative to a reference sequence. It also supports the visualization of synteny and multi-alignment data. Artemis is a genome browser and annotation tool that allows users to easily visualize, browse, and interpret large NGS datasets. It supports multiple sequence read views and variant displays, along with a comprehensive set of read alignment views and read alignment filters. It also has the ability to simultaneously display multiple different views of the same dataset to its users. Artemis can read EMBL and GENBANK database entries, FASTA sequence formats (indexed or raw), and other features in EMBL and GENBANK formats.

When reviewing an annotated genome (regardless of whether it is from a prokaryote or a eukaryote), it is always useful to randomly select a specific region and to use the chosen visualization/editing tools to carefully analyze the annotations together with the evidence provided. This evidence may include the *ab initio* predicted genes, the spliced RNA-seq alignments, or any homologous protein alignments. While browsing through the selected region, one may notice certain genes or clusters of genes that seem to contradict the displayed evidence. For instance, the RNA-seq data may support additional or different splice forms. Alternately, certain cross-species proteins may map to genomic regions where no gene has been previously predicted. Visual inspection can also reveal certain systematic problems with the annotation process, such as a tendency to miss genes with known database homologs or the appearance of repeats that overlap or mask many protein-coding genes. These problems may be addressed by changing parameter settings on the genome annotation pipeline, performing the necessary edits manually, or by choosing another tool. Multiple, iterative rounds of manual reviewing and manual editing followed by automated pipeline annotation are often necessary to complete a full and thorough genome annotation.

Summary

Genome annotation has evolved considerably over the past two decades. These changes have been driven, in part, by significant improvements in computational techniques (for gene prediction) and in part by a significant expansion in the number of known and annotated genomes

from an ever-growing number of diverse species. The availability of improved gene prediction tools, along with significantly expanded databases of well-annotated genes, proteins, and genomes, has moved genome annotation away from pure gene prediction to a more integrated, holistic approach that combines multiple lines of evidence to locate, identify, and functionally annotate genes. When combined with experimental data such as RNA-seq data or protein sequence data (from structural proteomics or expression-based proteomics), it is possible to obtain remarkably accurate and impressively complete annotations. This comprehensive blending of evidence is the basis for many newly developed, semi-automated or automated genome annotation pipelines and to many of the newer genome browsers and editors. However, not all genome annotation efforts can yield the same quantity or quality of information. Certainly prokaryotic genome annotation is faster, easier, and much more accurate than eukaryotic genome annotation. Indeed, the challenge of prokaryotic genome annotation is essentially a “solved problem,” while the challenge of eukaryotic genome annotation has to be considered as a “work in progress.”

Acknowledgments

The author thanks Andy Baxevanis and Roderic Guigó for their helpful comments and the use of material from prior editions of this book.

Internet Resources

Ab Initio Prokaryotic Gene Predictors

EasyGene (server)	www.cbs.dtu.dk/services/EasyGene
GeneMark.hmm (server)	opal.biology.gatech.edu/GeneMark/gmhmm.cgi
GeneMarkS (server)	opal.biology.gatech.edu/GeneMark/genemarks.cgi
GLIMMER (program)	www.cs.jhu.edu/~genomics/Glimmer
Prodigal (program)	github.com/hyatt/Prodigal

Ab Initio Eukaryotic Gene Predictors

GeneID (server)	genome.crg.es/geneid.html
GeneMark-ES (program)	opal.biology.gatech.edu/GeneMark
GeneZilla (program)	www.genezilla.org
GenomeScan (server)	hollywood.mit.edu/genomescan.html
GENSCAN (server)	hollywood.mit.edu/GENSCAN.html
HMMgene (server)	www.cbs.dtu.dk/services/HMMgene
SNAP (program)	korflab.ucdavis.edu/software.html

Hybrid/Extrinsic Eukaryotic Genome Finders

AUGUSTUS (server)	bioinf.uni-greifswald.de/augustus
AUGUSTUS-PPX (program)	bioinf.uni-greifswald.de/augustus
CONTRAST (program)	contra.stanford.edu/contrast
GeneID (server)	genome.crg.es/software/geneid
GeneWise (server)	www.ebi.ac.uk/Tools/psa/genewise
GenomeThreader (program)	genomethreader.org
GSNAP (program)	research-pub.gene.com/gmap
mGENE (program)	www.mgene.org

Hybrid/Extrinsic Eukaryotic Genome Finders

Mugsy-Annotator (program)	mugsy.sourceforge.net
SGP-2 (program)	genome.crg.es/software/sgp2
STAR (program)	code.google.com/archive/p/rna-star
Transomics (program)	linux1.softberry.com/berry.phtml?topic=transomics

tRNA and rRNA Finders

Rfam (server)	rfam.xfam.org
RNAmmer (server)	www.cbs.dtu.dk/services/RNAmmer
RNAMotif (program)	casegroup.rutgers.edu/casegr-sh-2.5.html
tRNAdb (server)	trnadb.bioinf.uni-leipzig.de/DataOutput/Welcome
tRNADB-CE (server)	trna.ie.niigata-u.ac.jp/cgi-bin/trnadb/index.cgi
tRNAfinder (server)	ei4web.yz.yamagata-u.ac.jp/~kinouchi/tRNAfinder
tRNAscan-SE (server)	lowelab.ucsc.edu/tRNAscan-SE

Phage-Finding Tools

Phage_Finder (program)	phage-finder.sourceforge.net
PHAST (server)	phast.wishartlab.com
PHASTER (server)	phaster.ca

Repeat Finding/Masking Tools

Dfam (server)	www.dfam.org
LTR_FINDER (server)	tlife.fudan.edu.cn/tlife/ltr_finder
LTRharvest (program)	genometools.org/index.html
MITE-Hunter (program)	target.iplantcollaborative.org/mite_hunter.html
Rebase (server)	www.girinst.org/rebase
RepeatMasker (program)	www.repeatmasker.org
RepeatScout (program)	bix.ucsd.edu/repeatscout
RetroPred (program)	www.juit.ac.in/attachments/RetroPred/home.html

Prokaryotic Genome Annotation Pipelines

BASys (server)	www.basys.ca
Prokka (program)	www.vicbioinformatics.com/software/prokka.shtml
RAST (server/program)	rast.nmpdr.org

Eukaryotic Genome Annotation Pipelines

BRAKER1 (program)	bioinf.uni-greifswald.de/bioinf/braker
EVM (program)	evidencemodeler.github.io
JIGSAW (program)	www.cbcb.umd.edu/software/jigsaw
MAKER2 (program)	www.yandell-lab.org/software/maker.html
PASA (program)	github.com/PASApipeline/PASApipeline/wiki

Genome Browsers and/or Editors

Artemis (program)	www.sanger.ac.uk/science/tools/artemis
Ensembl (program)	uswest.ensembl.org/downloads.html
GenomeView (program)	genomeview.org
JBrowse (program)	jbrowse.org
UCSC Genome Browser	hgdownload.cse.ucsc.edu/downloads.html
Web Apollo (program)	genomearchitect.github.io

Further Reading

- Hoff, K.J. and Stanke, M. (2015). Current methods for automated annotation of protein-coding genes. *Curr. Opin. Insect Sci.* 7, 8–14. A well-written and up-to-date summary of some of the latest developments in genome annotation with some very practical advice about which annotation tools should be used.
- Nielsen, P. and Krogh, A. (2005). Large-scale prokaryotic gene prediction and comparison to genome annotation. *Bioinformatics.* 21, 4322–4329. A very readable assessment of prokaryotic gene prediction and genome annotation.
- Yandell, M. and Ence, D. (2012). A beginner's guide to eukaryotic genome annotation. *Nat. Rev. Genet.* 13, 329–342. A nice, easy-to-read introduction to the processes involved in eukaryotic genome annotation along with useful descriptions of the available computational tools and best practices.
- Yoon, B. (2009). Hidden Markov models and their applications in biological sequence analysis. *Curr. Genomics* 10, 402–415. A comprehensive tutorial on HMMs that provides many useful examples and explanations of how different HMMs are constructed and used in gene prediction and gene sequence analysis.

References

- Abe, T., Inokuchi, H., Yamada, Y. et al. (2014). tRNADB-CE: tRNA gene database well-timed in the era of big sequence data. *Front. Genet.* 5: 114.
- Abeel, T., Van Parys, T., Saeys, Y. et al. (2012). GenomeView: a next-generation genome browser. *Nucleic Acids Res.* 40 (2): e12.
- Alexandersson, M., Cawley, S., and Patcher, L. (2003). SLAM: cross-species gene finding and alignment with a generalized pair hidden Markov model. *Genome Res.* 13: 496–502.
- Allen, J.E. and Salzberg, S.L. (2005). JIGSAW: integration of multiple sources of evidence for gene prediction. *Bioinformatics* 21: 3596–3603.
- Allen, J.E., Majoros, W.H., Pertea, M., and Salzberg, S.L. (2006). JIGSAW, GeneZilla, and GlimmerHMM: puzzling out the features of human genes in the ENCODE regions. *Genome Biol.* 7 (Suppl 1, S9): 1–13.
- Angiuoli, S.V., Dunning Hotopp, J.C., Salzberg, S.L., and Tettelin, H. (2011). Improving pan-genome annotation using whole genome multiple alignment. *BMC Bioinf* 12: 272.
- Arndt, D., Grant, J.R., Marcu, A. et al. (2016). PHASTER: a better, faster version of the PHAST phage search tool. *Nucleic Acids Res.* 44 (W1): W16–W21.
- Bao, Z. and Eddy, S.R. (2002). Automated de novo identification of repeat sequence families in sequenced genomes. *Genome Res.* 12: 1269–1276.
- Bellman, R.E. (1957). *Dynamic Programming*. Princeton: Princeton University Press.
- Besemer, J. and Borodovsky, M. (2005). GeneMark: web software for gene finding in prokaryotes, eukaryotes and viruses. *Nucleic Acids Res.* 33 (Web Server): W451–W454.
- Besemer, J., Lomsadze, A., and Borodovsky, M. (2001). GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions. *Nucleic Acids Res.* 29: 2607–2618.
- Birney, E. and Durbin, R. (1997). Dynamite: a flexible code generating language for dynamic programming methods used in sequence comparison. In: *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology, Halkidiki, Greece (21–26 June 1997)*, vol. 5, 56–64. Menlo Park, CA: AAAI Press.
- Birney, E., Clamp, M., and Durbin, R. (2004). GeneWise and Genomewise. *Genome Res.* 14: 988–995.
- Blanco, E., Parra, G., and Guigó, R. (2002). Using geneid to identify genes. In: *Current Protocols in Bioinformatics*, vol. 1, unit 4.3. New York: Wiley.

- Blattner, F.R., Plunkett, G. 3rd., Bloch, C.A. et al. (1997). The complete genome sequence of *Escherichia coli* K-12. *Science* 277: 1453–1462.
- Bobay, L.-M., Touchon, M., and Rocha, E.P.C. (2014). Pervasive domestication of defective prophages by bacteria. *Proc. Natl Acad. Sci. USA.* 111: 12127–12132.
- Borodovsky, M. and Lomsadze, A. (2011). Gene identification in prokaryotic genomes, phages, metagenomes, and EST sequences with GeneMarkS suite. *Curr. Protoc. Bioinformatics.* Chapter 4, Unit 4.5.1–17.
- Borodovsky, M. and McIninch, J. (1993). GeneMark: parallel gene recognition for both DNA strands. *Comput. Chem.* 17: 123–133.
- Borodovsky, M., Rudd, K.E., and Koonin, E.V. (1994). Intrinsic and extrinsic approaches for detecting genes in a bacterial genome. *Nucleic Acids Res.* 22: 4756–4767.
- Bose, M. and Barber, R.D. (2006). Prophage Finder: a prophage loci prediction tool for prokaryotic genome sequences. *In Silico Biol. (Gedrukt)* 6: 223–227.
- Burge, C. and Karlin, S. (1997). Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.* 268: 78–94.
- Burset, M. and Guigó, R. (1996). Evaluation of gene structure prediction programs. *Genomics.* 34: 353–357.
- Carver, T., Harris, S.R., Berriman, M. et al. (2012). Artemis: an integrated platform for visualization and analysis of high-throughput sequence-based experimental data. *Bioinformatics* 28: 464–469.
- Casjens, S. (2003). Prophages and bacterial genomics: what have we learned so far? *Mol. Microbiol.* 49: 277–300.
- Casper, J., Zweig, A.S., Villarreal, C. et al. (2018). The UCSC Genome Browser database: 2018 update. *Nucleic Acids Res.* 46 (D1): D762–D769.
- Coghlan, A., Fiedler, T.J., McKay, S.J. et al., and nGASP Consortium. (2008). nGASP – the nematode genome annotation assessment project. *BMC Bioinf* 9: 549.
- Cordaux, R. and Batzer, M.A. (2009). The impact of retrotransposons on human genome evolution. *Nat. Rev. Genet.* 10: 691–703.
- Delcher, A.L., Harmon, D., Kasif, S. et al. (1999). Improved microbial gene identification with GLIMMER. *Nucleic Acids Res.* 27: 4636–4641.
- Delcher, A.L., Bratke, K.A., Powers, E.C., and Salzberg, S.L. (2007). Identifying bacterial genes and endosymbiont DNA with Glimmer. *Bioinformatics* 23: 673–679.
- Dobin, A., Davis, C.A., Schlesinger, F. et al. (2013). STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* 29: 15–21.
- Dunham, I., Shimizu, N., Roe, B.A. et al. (1999). The DNA sequence of human chromosome 22. *Nature* 402: 489–495.
- Eddy, S.R. (2009). A new generation of homology search tools based on probabilistic inference. *Genome Inform.* 23: 205–211.
- Eilbeck, K., Moore, B., Holt, C., and Yandell, M. (2009). Quantitative measures for the management and comparison of annotated genomes. *BMC Bioinf* 10: 67.
- Ellinghaus, D., Kurtz, S., and Willhoeft, U. (2008). LTRharvest, an efficient and flexible software for de novo detection of LTR retrotransposons. *BMC Bioinf* 9: 18.
- Ezkurdia, I., Juan, D., Rodriguez, J.M. et al. (2014). Multiple evidence strands suggest that there may be as few as 19,000 human protein-coding genes. *Hum. Mol. Genet.* 23: 5866–5878.
- Fay, J.C. and Wu, C. (2003). Sequence divergence, functional constraint, and selection in protein evolution. *Annu. Rev. Genomics Hum. Genet.* 4: 213–235.
- Fernández-Suárez, X.M. and Schuster, M.K. (2010). Using the ensembl genome server to browse genomic sequence data. *Curr. Protoc. Bioinformatics.* Chapter 1, Unit 1.15.
- Fickett, J.W. and Tung, C.S. (1992). An assessment of protein coding measures. *Nucleic Acids Res.* 20: 6441–6450.
- Fouts, D.E. (2006). Phage_Finder: automated identification and classification of prophage regions in complete bacterial genome sequences. *Nucleic Acids Res.* 34: 5839–5851.
- Gelfand, M.S. (1995). Prediction of function in DNA sequence analysis. *J. Comput. Biol.* 2: 87–117.

- Gelfand, M.S. and Roytberg, M.A. (1993). Prediction of the exon-intron structure by a dynamic programming approach. *Biosystems*. 30: 173–182.
- Gelfand, M.S., Mironov, A.A., and Pevner, P.A. (1996). Gene recognition via spliced sequence alignment. *Proc. Natl. Acad. Sci. USA*. 93: 9061–9066.
- Gish, W. and States, D. (1993). Identification of protein coding regions by database similarity search. *Nat. Genet.* 3: 266–272.
- Grabherr, M.G., Haas, B.J., Yassour, M. et al. (2011). Full-length transcriptome assembly from RNA-seq data without a reference genome. *Nat. Biotechnol.* 29: 644–652.
- Gremme, G., Brendel, V., Sparks, M.E., and Kurtz, S. (2005). Engineering a software tool for gene structure prediction in higher organisms. *Inf. Software Technol.* 47: 965–978.
- Gross, S.S. and Brent, M.R. (2006). Using multiple alignments to improve gene prediction. *J. Comput. Biol.* 13: 379–393.
- Gross, S.S., Do, C.B., Sirota, M., and Batzoglou, S. (2007). CONTRAST: a discriminative, phylogeny-free approach to multiple informant de novo gene prediction. *Genome Biol.* 8: R269.
- Guigó, R. (1999). DNA composition, codon usage and exon prediction. In: *Genetic Databases* (ed. M. Bishop), 53–80. Cambridge, MA: Academic Press.
- Guigó, R. and Reese, M.G. (2005). EGASP: collaboration through competition to find human genes. *Nat. Methods* 2: 575–577.
- Guigó, R., Dermitzakis, E.T., Agarwal, P. et al. (2003). Comparison of mouse and human genomes followed by experimental verification yields an estimated 1,019 additional genes. *Proc. Natl. Acad. Sci. USA*. 100: 1140–1145.
- Guigó, R., Flicek, P., Abril, J.F. et al. (2006). EGASP: the human ENCODE genome annotation assessment project. *Genome Biol.* 7 (Suppl 1): S2.1–S2.31.
- Haas, B.J., Salzberg, S.L., Zhu, W. et al. (2008). Automated eukaryotic gene structure annotation using EVIDENCEModeler and the program to assemble spliced alignments. *Genome Biol.* 9: R7.
- Han, Y. and Wessler, S.R. (2010). MITE-Hunter: a program for discovering miniature inverted-repeat transposable elements from genomic sequences. *Nucleic Acids Res.* 38: e199.
- Harrow, J., Frankish, A., Gonzalez, J.M. et al. (2012). GENCODE: the reference human genome annotation for The ENCODE Project. *Genome Res.* 22: 1760–1774.
- Häsler, J. and Strub, K. (2006). Alu elements as regulators of gene expression. *Nucleic Acids Res.* 34: 5491–5497.
- Hoff, K.J. and Stanke, M. (2013). WebAUGUSTUS – a web service for training AUGUSTUS and predicting genes in eukaryotes. *Nucleic Acids Res.* 41 (Web Server issue): W123–W128.
- Hoff, K.J., Lange, S., Lomsadze, A. et al. (2016). BRAKER1: unsupervised RNA-seq-based genome annotation with GeneMark-ET and AUGUSTUS. *Bioinformatics* 32: 767–769.
- Holt, C. and Yandell, M. (2010). MAKER2: an annotation pipeline and genome-database management tool for second-generation genome projects. *BMC Bioinf* 12: 491.
- Hou, Y. and Lin, S. (2009). Distinct gene number–genome size relationships for eukaryotes and non-eukaryotes: gene content estimation for dinoflagellate genomes. *PLoS One* 4 (9): e6978.
- Hyatt, D., Chen, G.L., Locascio, P.F. et al. (2010). Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinf* 11: 119.
- Jühling, F., Mörl, M., Hartmann, R.K. et al. (2009). tRNAdb 2009: compilation of tRNA sequences and tRNA genes. *Nucleic Acids Res.* 37 (Database issue): D159–D162.
- Jurka, J., Kapitonov, V.V., Pavlicek, A. et al. (2005). Repbase update, a database of eukaryotic repetitive elements. *Cytogenet. Genome Res.* 110 (1–4): 462–467.
- Kalvari, I., Argasinska, J., Quinones-Olvera, N. et al. (2018). Rfam 13.0: shifting to a genome-centric resource for non-coding RNA families. *Nucleic Acids Res.* 46 (D1): D335–D342.
- Keller, O., Kollmar, M., Stanke, M., and Waack, S. (2011). A novel hybrid gene prediction method employing protein multiple sequence alignments. *Bioinformatics* 27: 757–763.
- Kent, W.J. (2002). BLAT – the BLAST-like alignment tool. *Genome Res.* 12: 656–664.
- Kim, D., Pertea, G., Trapnell, C. et al. (2013). TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biol.* 14: R36.

- Kinouchi, M. and Kuoakawa, K. (2006). tRNAfinder: a software system to find all tRNA genes in the DNA sequence based on the cloverleaf secondary structure. *J. Comput. Aided Chem.* 7: 116–126.
- König, S., Romoth, L.W., Gerischer, L., and Stanke, M. (2016). Simultaneous gene finding in multiple genomes. *Bioinformatics* 32: 3388–3395.
- Korf, I., Flicek, P., Duan, D., and Brent, M.R. (2001). Integrating genomic homology into gene structure prediction. *Bioinformatics*. 17: S140–S148.
- Kozak, M. (1987). An analysis of 5'-noncoding sequences from 699 vertebrate messenger RNAs. *Nucleic Acids Res.* 15: 8125–8148.
- Krogh, A. (1997). Two methods for improving performance of a HMM and their application for gene finding. In: *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology, Halkidiki, Greece (21–26 June 1997)*, vol. 5, 179–186. Menlo Park, CA: AAAI Press.
- Krogh, A., Mian, I.S., and Haussler, D. (1994). A hidden Markov model that finds genes in *E. coli* DNA. *Nucleic Acids Res.* 22: 4768–4678.
- Kulp, D., Haussler, D., Reese, M.G., and Eeckman, F.H. (1996). A generalized hidden Markov model for the recognition of human genes in DNA. In: *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology*, vol. 4, 134–142, June 12-15, 1996, St. Louis, MO. USA, AAAI Press, Menlo Park, California.
- Lagesen, K., Hallin, P., Rødland, E.A. et al. (2007). RNAmmer: consistent and rapid annotation of ribosomal RNA genes. *Nucleic Acids Res.* 35: 3100–3108.
- Lander, E.S., Linton, L.M., Birren, B. et al. (2001). Initial sequencing and analysis of the human genome. *Nature* 409: 860–921.
- Larsen, T.S. and Krogh, A. (2003). EasyGene – a prokaryotic gene finder that ranks ORFs by statistical significance. *BMC Bioinf* 4: 21.
- Lee, E., Helt, G.A., Reese, J.T. et al. (2013). Web Apollo: a web-based genomic annotation editing platform. *Genome Biol.* 14: R93.
- Li, W., Zhang, P., Fellers, J.P. et al. (2004). Sequence composition, organization, and evolution of the core Triticeae genome. *Plant J.* 40: 500–511.
- Lifton, R.P., Goldberg, M.L., Karp, R.W., and Hogness, D.S. (1978). The organization of the histone genes in *Drosophila melanogaster*: functional and evolutionary implications. *Cold Spring Harbor Symp. Quant. Biol.* 42: 1047–1051.
- Little, J.W. (2005). Lysogeny, prophage induction, and lysogenic conversion. In: *Phages: Their Role in Bacterial Pathogenesis and Biotechnology* (eds. M.K. Waldor, D.I. Friedman and S.L. Adhya), 37–54. Washington, DC: ASM Press.
- Lowe, T.M. and Eddy, S.R. (1997). tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res.* 25: 955–964.
- Lukashin, A.V. and Borodovsky, M. (1998). GeneMark.hmm: new solutions for gene finding. *Nucleic Acids Res.* 26: 1107–1115.
- Lunter, G. and Goodson, M. (2011). Stampy: a statistical algorithm for sensitive and fast mapping of Illumina sequence reads. *Genome Res.* 21: 936–939.
- Macke, T.J., Ecker, D.J., Gutell, R.R. et al. (2001). RNAMotif, an RNA secondary structure definition and search algorithm. *Nucleic Acids Res.* 29: 4724–4735.
- Meyer, I.M. and Durbin, R. (2002). Comparative ab initio prediction of gene structures using pair HMMs. *Bioinformatics* 18: 1309–1318.
- Naik, P.K., Mittal, V.K., and Gupta, S. (2008). RetroPred: a tool for prediction, classification and extraction of non-LTR retrotransposons (LINEs & SINEs) from the genome by integrating PALS, PILER, MEME and ANN. *Bioinformation* 2: 263–270.
- Overbeek, R., Olson, R., Pusch, G.D. et al. (2014). The SEED and the rapid annotation of microbial genomes using subsystems technology (RAST). *Nucleic Acids Res.* 42 (Database issue): D206–D214.
- Parra, G., Agarwal, P., Abril, J.F. et al. (2003). Comparative gene prediction in human and mouse. *Genome Res.* 13: 108–117.

- Pennisi, E. (2003). Bioinformatics. Gene counters struggle to get the right answer. *Science*. 301: 1040–1041.
- Perteau, M., Perteau, G.M., Antonescu, C.M. et al. (2015). StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. *Nat. Biotechnol.* 33: 290–295.
- Pribnow, D. (1975). Nucleotide sequence of an RNA polymerase binding site at an early T7 promoter. *Proc. Natl. Acad. Sci. USA*. 72: 784–788.
- Price, A.L., Jones, N.C., and Pevzner, P.A. (2005). De novo identification of repeat families in large genomes. *Bioinformatics* 21 (Suppl 1): i351–i358.
- Riley, M., Abe, T., Arnaud, M.B. et al. (2006). *Escherichia coli* K-12: a cooperatively developed annotation snapshot–2005. *Nucleic Acids Res.* 34: 1–9.
- Rogic, S., Mackworth, A.K., and Ouellette, F.B.F. (2001). Evaluation of gene-finding programs on mammalian sequences. *Genome Res.* 11: 817–832.
- Sakharkar, M., Passetti, F., de Souza, J.E. et al. (2002). ExInt: an exon intron database. *Nucleic Acids Res.* 30: 191–194.
- Sallet, E., Gouzy, J., and Schiex, T. (2014). EuGene-PP: a next-generation automated annotation pipeline for prokaryotic genomes. *Bioinformatics* 30: 2659–2661.
- Schweikert, G., Behr, J., Zien, A. et al. (2009). mGene.web: a web service for accurate computational gene finding. *Nucleic Acids Res.* 37 (Web Server issue): W312–W316.
- Seemann, T. (2014). Prokka: rapid prokaryotic genome annotation. *Bioinformatics* 30: 2068–2069.
- Shine, J. and Dalgarno, L. (1975). Determinant of cistron specificity in bacterial ribosomes. *Nature* 254: 34–38.
- Simão, F.A., Waterhouse, R.M., Ioannidis, P. et al. (2015). BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs. *Bioinformatics*. 31: 3210–3212.
- Slater, G.S. and Birney, E. (2005). Automated generation of heuristics for biological sequence comparison. *BMC Bioinf* 6: 31.
- Slupska, M.M., King, A.G., Fitz-Gibbon, S. et al. (2001). Leaderless transcripts of the crenarchaeal hyperthermophile *Pyrobaculum aerophilum*. *J. Mol. Biol.* 309: 347–360.
- Souvorov, A., Kapustin, Y., Kiryutin, B. et al. (2010). Gnomon – NCBI eukaryotic gene prediction tool. *Natl Cent. Biotechnol. Inf.* 2010: 1–24.
- Sperisen, P., Iseli, C., Pagni, M. et al. (2004). Trome, trEST and trGEN: databases of predicted protein sequences. *Nucleic Acids Res.* 32 (Database issue): D509–D511.
- Steijger, T., Abril, J.F., Engström, P.G. et al., and RGASP Consortium (2013). Assessment of transcript reconstruction methods for RNA-seq. *Nat. Methods* 10: 1177–1184.
- Stothard, P. and Wishart, D.S. (2005). Circular genome visualization and exploration using CGView. *Bioinformatics* 21: 537–539.
- Subramanian, S., Mishra, R.K., and Singh, L. (2003). Genome-wide analysis of microsatellite repeats in humans: their abundance and density in specific genomic regions. *Genome Biol.* 4: R13.
- Tarailo-Graovac, M. and Chen, N. (2009). Using RepeatMasker to identify repetitive elements in genomic sequences. *Curr. Protoc Bioinformatics*. Chapter 4, Unit 4.10.
- Taruscio, D. and Mantovani, A. (2004). Factors regulating endogenous retroviral sequences in human and mouse. *Cytogenet. Genome Res.* 105: 351–362.
- Thibaud-Nissen, F., DiCuccio, M., Hlavina, W. et al. (2016). The NCBI eukaryotic genome annotation pipeline. *J. Anim. Sci.* 94 (Suppl 4): 184.
- Trapnell, C., Pachter, L., and Salzberg, S.L. (2009). TopHat: discovering splice junctions with RNA-seq. *Bioinformatics* 25: 1105–1111.
- Trapnell, C., Roberts, A., Goff, L. et al. (2012). Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and cufflinks. *Nat. Protoc.* 7: 562–578.
- Van Domselaar, G.H., Stothard, P., Shrivastava, S. et al. (2005). BASys: a web server for automated bacterial genome annotation. *Nucleic Acids Res.* 33 (Web Server issue): W455–W459.
- Wang, Z., Gerstein, M., and Snyder, M. (2009). RNA-seq: a revolutionary tool for transcriptomics. *Nat. Rev. Genet.* 10: 57–63.

- Waterhouse, R.M., Tegenfeldt, F., Li, J. et al. (2013). OrthoDB: a hierarchical catalog of animal, fungal and bacterial orthologs. *Nucleic Acids Res.* 41 (Database issue): D358–D365.
- Wegrzyn, J.L., Liechty, J.D., Stevens, K.A. et al. (2014). Unique features of the loblolly pine (*Pinus taeda* L.) megagenome revealed through sequence annotation. *Genetics* 196: 891–909.
- Westesson, O., Skinner, M., and Holmes, I. (2013). Visualizing next-generation sequencing data with JBrowse. *Briefings Bioinf.* 14: 172–177.
- Wheeler, T.J., Clements, J., Eddy, S.R. et al. (2013). Dfam: a database of repetitive DNA based on profile hidden Markov models. *Nucleic Acids Res.* 41 (Database issue): D70–D82.
- Will, C.L. and Lührmann, R. (2011). Spliceosome structure and function. *Cold Spring Harbor Perspect. Biol.* 3 (7), pii: a003707.
- Winsor, G.L., Lo, R., Ho Sui, S.J. et al. (2005). Pseudomonas aeruginosa genome database and PseudoCAP: facilitating community-based, continually updated, genome annotation. *Nucleic Acids Res.* 33 (Database issue): D338–D343.
- Wootton, J.C. and Federhen, S. (1993). Statistics of local complexity in amino acid sequences and sequence databases. *Comput. Chem.* 17: 149–163.
- Wu, T.D., Reeder, J., Lawrence, M. et al. (2016). GMAP and GSNAP for genomic sequence alignment: enhancements to speed, accuracy, and functionality. *Methods Mol. Biol.* 1418: 283–334.
- Xu, Z. and Wang, H. (2007). LTR_FINDER: an efficient tool for the prediction of full-length LTR retrotransposons. *Nucleic Acids Res.* 35 (Web Server issue): W265–W268.
- Yeh, R., Lim, L.P., and Burge, C. (2001). Computational inference of the homologous gene structures in the human genome. *Genome Res.* 11: 803–816.
- Zhang, M.Q. (2002). Computational prediction of eukaryotic protein coding genes. *Nat. Rev. Genet.* 3: 698–709.
- Zhou, Y., Liang, Y., Lynch, K.H. et al. (2011). PHAST: a fast phage search tool. *Nucleic Acids Res.* 39 (Web Server issue): W347–W352.
- Zhu, H., Hu, G., Yang, Y. et al. (2007). MED: a new non-supervised gene prediction algorithm for bacterial and archaeal genomes. *BMC Bioinf* 8: 97.

6

Predictive Methods Using RNA Sequences

Michael F. Sloma, Michael Zuker, and David H. Mathews

Introduction

RNA is a versatile biopolymer that plays many roles beyond simply carrying and recognizing genetic information as messenger RNA (mRNA) and transfer RNA (tRNA), respectively. It has been known for decades that RNA sequences can catalyze RNA cleavage and ligation (Doudna and Cech 2002) and that RNA is an important component of the signal recognition particle (SRP) (Walter and Blobel 1982) that directs the export of proteins out of the cell. More recently, additional roles for RNA have been discovered. Ribosomal RNAs (rRNAs) catalyze peptide bond formation during protein synthesis (Nissen et al. 2000; Hansen et al. 2002), small nuclear RNAs (snRNAs) and self-splicing introns catalyze pre-mRNA splicing reactions, microRNAs (miRNAs) and small interfering RNAs (siRNA) regulate gene expression by binding to mRNAs, and mRNAs regulate their own expression by binding metabolites via RNA structures called riboswitches. RNA plays roles in other crucial processes including development (Lagos-Quintana et al. 2001; Lau et al. 2001) and the immune system (Cullen 2002). Furthermore, RNA can be made to evolve *in vitro* to catalyze reactions that do not occur in nature (Bittker et al. 2002).

RNA is also an important target and agent for the pharmaceutical industry. In the ribosome, RNA is the target of several classes of antibiotics. mRNA is the target of drugs that work on the antisense principle (Dias and Stein 2002) or by RNA interference, also known as RNAi (Castanotto and Rossi 2009). Recent work has shown that RNA can be targeted specifically by small molecules (Disney et al. 2016).

To fully understand the mechanism of action or to target an RNA sequence, the structure of the RNA under investigation needs to be understood. RNA structure has three levels of organization, as shown in Figure 6.1 (Tinoco and Bustamante 1999). The first level – the primary structure (Figure 6.1a) – is simply the linear sequence of nucleotides in the RNA molecule. The secondary structure (Figure 6.1b) is defined by the base-pairing interactions (both Watson–Crick pairs and G–U pairs) that take place within the RNA polymer. Finally, the tertiary structure (Figure 6.1c) is the three-dimensional arrangement of the atoms in the RNA sequence and, therefore, includes all the non-canonical contacts.

Often, the secondary structure of an RNA sequence is solved before its tertiary structure because there are accurate experimental and computational methods for determining the secondary structure of an RNA sequence and because knowledge of the secondary structure is often helpful in designing constructs for tertiary structure determination. A typical RNA secondary structure, illustrated in Figure 6.2, is composed of both helical and loop regions. The helical regions are composed of canonical base pairs. The loop regions take different forms, depending on the number of closing base pairs and the distribution of unpaired nucleotides. They can be hairpin loops, in which the backbone makes a 180° bend; internal loops, in which a helix is interrupted with two strands of unpaired nucleotides; bulge loops, in which a helix is interrupted with a single strand of unpaired nucleotides; and multibranch loops (also called

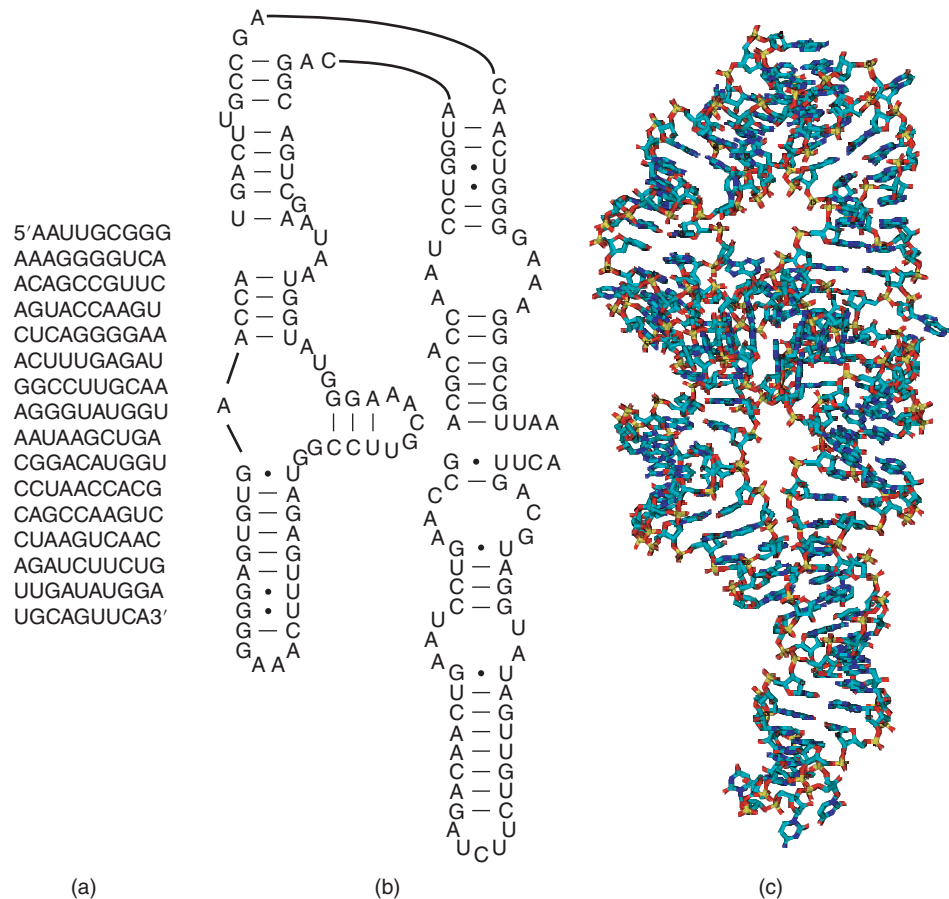


Figure 6.1 The three levels of organization of RNA structure. (a) The primary sequence, (b) the secondary structure (Cannone et al. 2002), and (c) the tertiary structure (Cate et al. 1996) of a domain of the group I intron from *Tetrahymena*. The secondary structure illustrates the canonical base pairs and the tertiary structure captures the three-dimensional arrangement of atoms. Reproduced with permission of AAAS.

helical junctions), from which more than two helices exit. Although secondary structure diagrams often do not explicitly illustrate nucleotide interactions in loop regions, these regions are responsible for the non-canonical interactions that stabilize the structure.

In the absence of a tertiary structure, the “gold standard” for predicting the placement of loops and helices is comparative sequence analysis, which uses evolutionary evidence found in sequence alignments to determine base pairs (Pace et al. 1999) (see also Chapter 8 for information on multiple sequence alignment methods). Base pairs predicted by comparative sequence analysis for large (LSU) and small subunit (SSU) rRNA were 97% accurate when compared with high-resolution crystal structures (Gutell et al. 2002).

RNA structure prediction is a large field, with hundreds of computational tools available for predicting the structure of an RNA molecule. This chapter presents some of the most popular methods used to deduce RNA secondary structure based on sequence. This chapter also presents ways in which additional information can be used to improve structure prediction accuracy, including methods that find a common structure for multiple homologous sequences and methods that use experimental data. To that end, RNA folding thermodynamics and dynamic programming are introduced. The Mfold and RNAstructure web servers, commonly used tools for predicting RNA secondary structure, are described in detail. Alternative software tools are also mentioned. This chapter concludes with a brief introduction to the methods used for RNA tertiary structure prediction. Additional resources with more in-depth information on specific tools are provided for the interested reader.

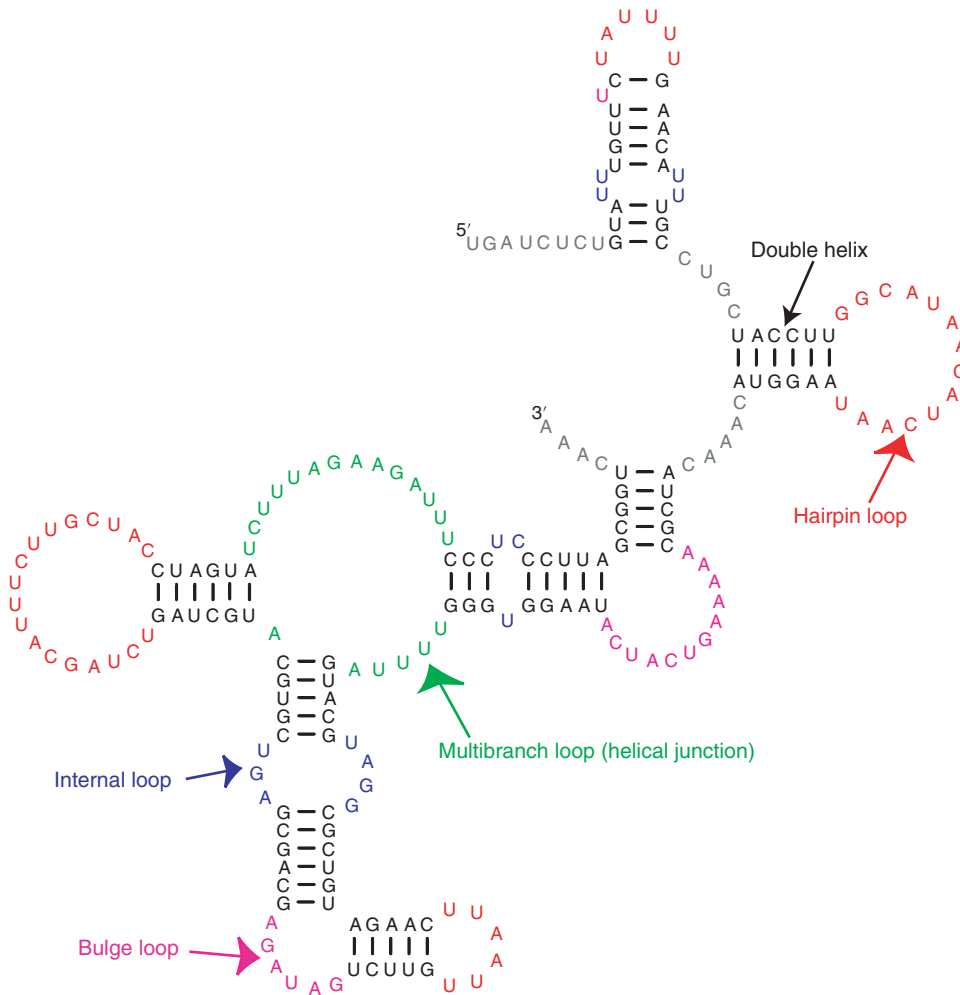


Figure 6.2 The RNA secondary structure of the 3' untranslated region of the *Drosophila sucinea* R2 element (Lathe and Eickbush 1997; Mathews et al. 1997). Base pairs in non-helical regions, known as loops, are colored by type of loop, which is labeled.

Overview of RNA Secondary Structure Prediction Using Thermodynamics

Many methods for RNA secondary structure prediction rely on a nearest neighbor model for predicting the stability of an RNA secondary structure, in terms of the Gibbs free energy change at 37 °C (ΔG°_{37}) (Box 6.1) (Xia et al. 1998, 1999; Mathews et al. 1999a, 2004; Turner 2000; Turner and Mathews 2010). The rules for predicting stability use a nearest neighbor model because the stability of each base pair depends only on the most adjacent pairs and the total free energy is the sum of each contribution. In-depth reviews of the determination of nearest neighbor parameters from experiments are available (Schroeder and Turner 2009; Andronescu et al. 2014).

Box 6.1 Gibbs Free Energy

The Gibbs free energy of formation for an RNA structure (ΔG°) quantifies the equilibrium stability of that structure at a specific temperature. For example, consider an RNA structure A that is at equilibrium with the random-coil (i.e. unstructured) conformation.

(Continued)

Box 6.1 (Continued)

The relative concentration of each conformation is governed by the equilibrium constant, K_{eq} , as illustrated in Figure 6.3a. K_{eq} is related to the Gibbs free energy by the relationship:

$$K_{\text{eq}} = \frac{[\text{Conformation A}]}{[\text{Random coil}]} = e^{-\Delta G^\circ / RT} \quad (6.1)$$

where R is the gas constant ($1.987 \text{ cal mol}^{-1} \text{ K}^{-1}$) and T is the absolute temperature (in Kelvin).

Furthermore, for multiple alternative conformations, A and B , for which there is an equilibrium distribution of conformations, K'_{eq} , as shown in Figure 6.3b, describes the distribution of strands between the structures. In this case, the free energy of each conformation relative to the random coil also describes the population of each conformation:

$$K'_{\text{eq}} = \frac{[\text{Conformation A}]}{[\text{Conformation B}]} = e^{-(\Delta G_A^\circ - \Delta G_B^\circ) / RT} \quad (6.2)$$

This generalizes to any number of conformations. Therefore, the lowest free energy conformation is the most probable conformation for an RNA molecule at equilibrium. This is commonly called the minimum free energy structure.

Free energies are expressed in units of joules per mole (J mol^{-1}) in SI units. Commonly, for RNA folding stability, these are still frequently expressed in units of kilocalorie per mole (kcal mol^{-1}), where a calorie = 4.184 J . A difference in the Gibbs free energy change of $1.42 \text{ kcal mol}^{-1}$ at 37°C (human body temperature; 310.15 K) changes the equilibrium constant by a factor of 10, which can be shown by plugging $1.42 \text{ kcal mol}^{-1}$ in for $\Delta G_A^\circ - \Delta G_B^\circ$ in Eq. (6.2).

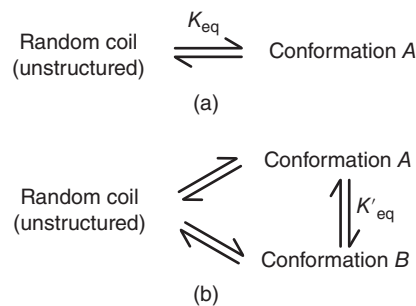


Figure 6.3 An illustration of the equilibria of RNA structures in solution. (a) The equilibrium between conformation A and the random coil structure. K_{eq} , which is related to the standard state free energy change at 37°C (ΔG_{37}°), describes the equilibrium. (b) The equilibrium between two conformations, A and B , and the random coil. K'_{eq} , which is related to the free energy of folding for both A and B , describes the population of conformation A versus conformation B .

An example of a nearest neighbor stability calculation is shown in Figure 6.4. Terms for helical stacking, loop initiation, and unpaired nucleotide stacking contribute to the total conformational free energy. Favorable free energy increments are always less than zero. The free energy increments of base pairs are counted as stacks of adjacent pairs. The consecutive CG base pairs, for example, contribute $-3.3 \text{ kcal mol}^{-1}$ (Xia et al. 1998). Note that the loop regions have

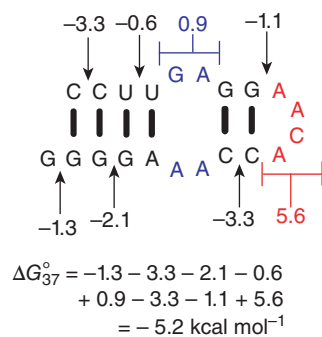


Figure 6.4 Prediction of conformational free energy for a conformation of RNA sequence CCUUGAGGAACACAAAGGGG. Each contributing free energy increment is labeled. The total free energy is the sum of the increments. For this estimated stability of $-5.2 \text{ kcal mol}^{-1}$, there is a population of 4600 folded strands to every one unfolded ($K_{\text{eq}} = 4600$; Box 6.1).

unfavorable increments called loop initiation energies that largely reflect an entropic cost for constraining the nucleotides in the loop. For example, the hairpin loop of four nucleotides has an initiation free energy change of $5.6 \text{ kcal mol}^{-1}$ (Mathews et al. 2004). Unpaired nucleotides in loops can provide favorable energy increments either as stacked nucleotides or as mismatched pairs. The 3'-most G shown in Figure 6.4, called a dangling end, stacks on the terminal base pair and provides $-1.3 \text{ kcal mol}^{-1}$ of stability. The first mismatch in the hairpin loop with this sequence context contributes $-1.1 \text{ kcal mol}^{-1}$ of stability.

The nearest neighbor free energy parameters utilize sequence-dependent terms for predicting the free energy increments of loop regions (Mathews et al. 1999a) to reflect experimental observations. For example, the 2×2 internal loop (an internal loop with two unpaired nucleotides on each side of the loop) can vary in stability from -2.6 to $+2.8 \text{ kcal mol}^{-1}$ depending on the sequence of the closing pair and mismatches (Schroeder et al. 1999).

The structure with the lowest ΔG° is the most likely structure at equilibrium, but this is not everything that there is to know about the equilibrium of structures. Another useful quantity, the partition function Q , provides a description of the structural ensemble (denoted as s , the set of all structures that the RNA molecule can adopt) of an RNA molecule by summing the equilibrium constant of each conformation:

$$Q = \sum_s e^{-\Delta G^\circ / RT} \quad (6.3)$$

The partition function can be used to calculate the probability that an RNA molecule in the ensemble of structures adopts conformation A :

$$P(A) = \frac{e^{-\Delta G_A^\circ / RT}}{Q} \quad (6.4)$$

When predicting structures, the structure with the lowest ΔG° can have a low probability. However, many of the low free energy structures will contain the same base pairs, and those pairs can thus have a high probability of forming. The probability of a specific base pair forming in an RNA structure is given by:

$$P(i \text{ paired to } j) = \frac{1}{Q} \sum_{s' \in s_{ij}} e^{-\Delta G_{s'}^\circ / RT} \quad (6.5)$$

where s_{ij} is the set of structures in which the nucleotide at index i is paired to the nucleotide at index j .

Partition functions are the basis for many computational methods to study RNA structure, including methods to identify a common structure for multiple sequences (Harmanci et al. 2011; Will et al. 2012), to identify mutations that alter RNA structure (Halvorsen et al. 2010; Sabarinathan et al. 2013; Salari et al. 2013), and to estimate accessibility to oligonucleotide binding (Lu and Mathews 2007; Tafer et al. 2008).

Dynamic Programming

In the previous section, thermodynamic rules were introduced for RNA secondary structure prediction that require searching the full space of possible structures, either to find the best scoring structure or to calculate the equilibrium constant for each structure. How is this search performed? The naive approach would be to explicitly generate every possible conformation, evaluate the free energy of each, and then choose the conformation that had the lowest (best) free energy.

One estimate is that there are $(1.8)^N$ secondary structures possible for a sequence of N nucleotides (Zuker and Sankoff 1984) – that is 3×10^{25} structures for a modest length sequence of 100 nucleotides. Given a fast computer that can calculate the free energy for 10 000 structures in a second, this approach would require 1.6×10^{14} years! Clearly, a better solution needs to be implemented for a problem of this size.

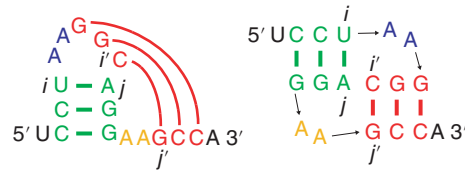


Figure 6.5 A simple RNA pseudoknot. This figure illustrates two representations of the same simple, H-type pseudoknot. A pseudoknot is defined by two base pairs such that $i-j$ and $i'-j'$ are two pairs with ordering $i < i' < j < j'$. The base pair between nucleotides i and j defines an enclosed region. The base pair between nucleotides i' and j' spans the enclosed region and an adjacent region, making the pseudoknot.

The most commonly employed solution for cases such as this is called dynamic programming, which uses recursion and tabulation of intermediate results to accelerate the calculation (Nussinov and Jacobson 1980; Zuker and Stiegler 1981). Appendix 6.A describes this method in detail for the interested reader. The dynamic programming approach can find both the minimum free energy structure and the partition function much faster than brute force, with an asymptotic performance that scales with computational time by N^3 (denoted as $O(N^3)$) and $O(N^2)$ in data storage for sequences of length N when pseudoknots are excluded from the calculation (Box 6.2). A pseudoknot, illustrated in Figure 6.5, occurs when there are non-nested base pairs (Liu et al. 2010). For example, the simplest pseudoknot occurs when there are two base pairs $i-j$ and $i'-j'$ such that $i < i' < j < j'$. It had been assumed that pseudoknots could not be predicted by a polynomial time dynamic programming until Rivas and Eddy (1999) presented a polynomial time dynamic programming algorithm that can predict structures containing a certain class of pseudoknots that is sufficiently rich to cover all cases of practical importance. However, their algorithm is $O(N^6)$ in time and $O(N^4)$ in storage, making the calculation impractical for sequences longer than about 300 nucleotides (Rivas and Eddy 1999; Condon et al. 2004). Other adaptations of the algorithm have improved scaling, but are still limited in practice to the sequence length on which they can be applied (Reeder and Giegerich 2004). A partition function algorithm that includes pseudoknots (Dirks and Pierce 2003) is $O(N^5)$ in time and $O(N^4)$ in storage, and is likewise only useful for sequences up to 200 nucleotides in length.

Box 6.2 Algorithm Complexity

In computer science, algorithm complexity describes the scaling of a calculation in the worst case scenario. It is expressed using the “big- O ” notation, which can be read as “order.” Algorithms that are $O(N)$ in time require a linear increase in computational time as the size parameter, N , increases. $O(N^2)$ and $O(N^3)$ algorithms scale by the square and cube of the parameter N , respectively. Therefore, the dynamic programming algorithm for RNA secondary structure prediction, which is $O(N^3)$, where N is the number of nucleotides, requires roughly eight times the execution time for a sequence twice as long. This is a fairly expensive calculation, as compared with sorting a list, which can generally be accomplished in $O(N \log(N))$ time.

The big- O notation also applies to the scaling of memory (also called storage) used by an algorithm. Secondary structure prediction requires two-dimensional arrays of size $N \times N$. Therefore, in storage, the secondary structure prediction algorithm is $O(N^2)$.

Variants of the dynamic programming algorithm for finding the minimum free energy structure (Mathews et al. 1999a; Wuchty et al. 1999) can also predict structures with free energy greater than the lowest free energy structure, using an additional constant factor of time and memory. These are called suboptimal structures (Zuker 1989). Suboptimal structures with estimated free energy changes close to the lowest free energy structure provide important alternative hypotheses for the actual structure. This is due to the fact that nearest neighbor parameters are imperfect; also, ignoring motifs like pseudoknots can result in a suboptimal structure being more accurate than the lowest free energy structure.

Accuracy of RNA Secondary Structure Prediction

The accuracy of RNA secondary structure prediction can be assessed by predicting structures for RNA sequences with known secondary structures. For a collection of structures assembled to test prediction accuracy, including SSU rRNA (Cannone et al. 2002), LSU rRNA (Cannone et al. 2002), 5S rRNA (Szymanski et al. 2000), group I introns (Cannone et al. 2002), group II introns (Michel et al. 1989), RNase P RNA (Brown 1999), SRP RNA (Larsen et al. 1998), and tRNA (Sprinzl et al. 1998), 73% of base pairs in the known structure were, on average, correctly predicted (Mathews et al. 2004). For these calculations, the SSU and LSU rRNAs are divided into domains of fewer than 700 nucleotides based on the known secondary structure (Mathews et al. 1999a). Although this level of accuracy is sufficient to make hypotheses about a structure of interest, a more accurate prediction is often desirable. There are two general approaches to improving the accuracy of a secondary structure prediction, both of which try to reduce the number of incorrect structures that are considered in the search step. One approach is to use low-resolution experimental data (Sloma and Mathews 2015) and the other is to predict a structure common to multiple homologs (Seetin and Mathews 2012a).

Experimental Methods to Refine Secondary Structure Prediction

Low-resolution experimental methods use either enzymatic cleavage or chemical modification reagents that preferentially react with either double-stranded or single-stranded nucleotides. Examining the reactivity at each position identifies which nucleotides are in stems and which are in loops, but gives no information on what the pairing partners of the double-stranded nucleotides are. Commonly used reagents for these experiments include RNase V1, which cleaves RNA molecules in double-stranded regions; RNase T, which cleaves RNA molecules after an unpaired guanine nucleotide, and RNase T2, which cleaves after an unpaired nucleotide of any type; dimethyl sulfate, which modifies unpaired adenine and cytosine nucleotides; and selective 2'-hydroxyl acylation analyzed by primer extension (SHAPE) reagents, a group of chemicals that modify any unpaired nucleotide. Recent advances have allowed these methods to be used in concert with massively parallel sequencing to probe the secondary structure of many different RNA molecules simultaneously inside living cells (Spitale et al. 2013; Ding et al. 2014; Rouskin et al. 2014; Talkish et al. 2014).

This experimental information can be used in an RNA secondary structure prediction algorithm in one of two ways. One is to forbid the search step of the dynamic programming algorithm from considering any structures that are inconsistent with the experimental data. This often dramatically increases the accuracy for sequences that are poorly predicted without experimental data. For example, for the 5S rRNA sequence from *Escherichia coli*, which is poorly predicted without experimental constraints, the accuracy improves from 26% to 87% using enzymatic cleavage data (Speek and Lind 1982; Mathews et al. 1999a; Szymanski et al. 2000).

Another way to apply experimental data to improve the prediction is to assign a pseudo-free energy penalty to structures that do not precisely match the data, rather than forbidding them entirely. This approach is useful because, while reactivity to enzymes or chemical probes is strongly correlated with single-strandedness, some double-stranded nucleotides can still be highly reactive (Sukosd et al. 2013). In these cases, using a soft restraint where inconsistent structures are merely penalized (instead of forbidden) allows prediction of structures that are “mostly consistent” with the data. This approach was originally used for SHAPE (Deigan et al. 2009) and has also been applied to both dimethyl sulfate modification (Cordero et al. 2012) and enzymatic cleavage (Underwood et al. 2010) experimental data.

Predicting the Secondary Structure Common to Multiple RNA Sequences

An alternative approach to improving RNA secondary structure prediction is to make use of information provided by evolution. Homologous RNA molecules that perform the same

function in different organisms are expected to form similar structures, even though their sequences may have diverged substantially. In particular, compensatory mutations, in which a mutation at one position would disrupt a base pair but a second mutation at its partner position restores base pairing, is strong evidence that a base pair exists. Algorithms can automate this process by restricting their search space to structures that can be adopted by all of the homologs, or by weighting more heavily structures that contain compensatory mutations.

The basis of comparative sequence analysis is the detection of conserved structure, as inferred from sequence differences between species or between sequences discovered by *in vitro* evolution (Pace et al. 1999). The assumption of a conserved secondary structure eliminates from consideration the many possible secondary structures for a single sequence that the set of sequences across evolution cannot all adopt. In other words, the multiple sequences constrain the possible secondary structure. These constraints can also be used as auxiliary information in the prediction of secondary structure. Manual comparative sequence analysis can be highly accurate, with over 97% of base pairs inferred for rRNAs present in subsequently solved crystal structures (Gutell et al. 2002), but it requires significant skill and effort. Computer algorithms that automate these comparative analyses are still not as accurate as manual comparative analysis in which the models are refined over time.

RNA secondary structure prediction algorithms that incorporate information from multiple sequences can be divided between those that are constrained by an initial sequence alignment and those that are not. In general, those methods that are constrained by an initial alignment are not as robust because of the limitations in the alignment, but are computationally faster.

Algorithms That Are Constrained by an Initial Alignment

Several programs have been developed for finding the secondary structure common to a set of aligned sequences (Lück et al. 1996, 1999; Juan and Wilson 1999; Hofacker et al. 2002). A popular approach, called Alifold, uses a sequence alignment to constrain secondary structure prediction by free energy minimization or constraining the calculation of the partition function (Hofacker et al. 2002; Bernhart et al. 2008). Additional energy terms are added to the conformation free energy to favor compensating base changes and sequence conservation. This program is available as part of the Vienna RNA Package (Lorenz et al. 2011) and as a web server.

Another approach for finding a structure common to multiple sequences, called Pfold, uses a stochastic context-free grammar (Knudsen and Hein 1999). The grammar defines rules for generating a sequence together with a secondary structure. These rules, encoded as probability parameters, are estimated from a sequence alignment and known, common secondary structures of a number of tRNAs and LSU rRNAs. These sequences and structures are referred to as the *training set*. A given sequence is folded using a dynamic programming algorithm that determines a structure with a maximum probability of being generated by the stochastic context-free grammar.

Algorithms That Are Not Constrained by the Initial Alignment

Dynamic programming can be used to simultaneously predict the sequence alignment and common secondary structure for multiple RNA sequences (Sankoff 1985). In general, this approach is $O(N_1^3 N_2^3 N_3^3 \dots)$ in time, where N_1 is the length of the first sequence, N_2 the length of the second sequence, and so forth, making it computationally impractical. Two programs are available that are based on this approach: FoldAlign (Havgaard et al. 2005) and Dynalign (Mathews and Turner 2002; Fu et al. 2014). Given the time required to run them, these programs are limited to two sequences, but both have been extended using pairwise calculations to work on more than two sequences (Torarinsson et al. 2007; Xu and Mathews 2011). These programs are the best choices to predict structures when the set of sequences is highly diverged.

An alternative approach to predicting RNA secondary structure without an input alignment is to fold the sequences, align sequences, and then combine the information to predict a conserved secondary structure. Three modern tools that take this approach are LocARNA (Will et al. 2007, 2012), PARTS (Harmanci et al. 2008, 2009), and TurboFold (Harmanci et al. 2011). These programs are faster than Dynalign and FoldAlign, but require higher sequence identity to work well.

Practical Introduction to Single-Sequence Methods

This section introduces two web servers: the Mfold web server and the RNAstructure web server. Both web servers provide structure predictions of similar accuracy for single sequences, and either server can be used depending upon the features desired. The Mfold server additionally provides an interface to simulate the melting of bimolecular complexes (Dimitrov and Zuker 2004). The RNAstructure web server additionally provides methods for determining conserved secondary structures in multiple homologs, siRNA design, and bimolecular structure prediction (Bellaousov et al. 2013).

Using the Mfold Web Server

Mfold is an RNA secondary structure prediction package available both as a web server and as code for compilation on Unix/Linux machines (Mathews et al. 1999a; Zuker 2003). It uses a set of nearest neighbor parameters for free energies at 37 °C (Mathews et al. 1999a). Minimum free energy and suboptimal secondary structures, generated heuristically (Zuker 1989), are predicted. Suboptimal structures represent alternative structures to the lowest free energy structure and reflect both the possibility that an RNA sequence may have more than a single structure (Schultes and Bartel 2000) and the fact that the energy rules contain some uncertainty (Mathews et al. 1999a; Layton and Bundschuh 2005; Zuber et al. 2017). Mfold also predicts energy dot plots, which display the lowest free energy conformation possible for each possible base pair (Zuker and Jacobson 1995). These plots conveniently demonstrate all possible base pairs within a user-specified increment of the lowest free energy structure and predicted structures can be color annotated to demonstrate regions in the structure for which many folding alternatives exist (Zuker and Jacobson 1998).

Figure 6.6a,b shows the top and bottom of the input form on the Mfold server, respectively. A sequence name can be entered in the box labeled *Enter sequence name* and the sequence is typed (or pasted from the clipboard) in the box labeled *Enter the sequence to be folded in the box below*. As the caption explains, non-alphabetic characters are ignored and do not interfere with sequence interpretation. For example, the form shows a tRNA sequence called RD1140 (Sprinzl et al. 1998) pasted into the sequence field. The remainder of the form has default values that can be changed by advanced users. The next box provides the option of constraining structure prediction with auxiliary evidence derived from enzymatic cleavage experiments (Knapp 1989), comparative sequence analysis (Pace et al. 1999), or biological intuition. Next, the default is for linear RNA sequence folding, although circular sequences can also be folded by changing the option from *linear* to *circular*. Note that the folding temperature is fixed at 37 °C using the current parameters. An older, less complete set of parameters allows secondary structure prediction at other temperatures (Jaeger et al. 1989), but it is recommended that the current parameters be used for most applications. The older parameters can be used for folding by following the link at the top of the page to the *RNA mfold version 2.3* server (not shown in Figure 6.6). The percent suboptimality number (5 by default) is the maximum percent difference in free energy from the lowest free energy structure that is allowed when generating suboptimal secondary structures. The upper bound on the computed foldings, default at 50, is the maximum number of suboptimal secondary structures to be predicted. The window parameter controls how different each suboptimal structure must be from all others. It defaults to a

Applications

- RNA Folding Form
- DNA Folding Form
- Structure Display and Free Energy Determination
- RNA Folding Form (version 2.3 server)

View Folding Results

- Folding Results

Documentation

- Mfold References
- FAQs
- Folding & output options
- Folding with constraints

Software

- Mfold

About

- About

Contact

- Contact

RNA Folding Form

M. Zuker
 Mfold web server for nucleic acid folding and hybridization prediction.
Nucleic Acids Res. **31** (13), 3406-15, (2003)
[\[Abstract\]](#) [\[Full Text\]](#) [\[Supplementary Material\]](#) [\[Additional Information\]](#)

The folding temperature is fixed at 37°. You may still fold with the older version 2.3 RNA parameters, which allow the temperature to be varied.
[DNA mfold server](#) [Quickfold](#). Fold many short RNA or DNA sequences at once.

Enter sequence name:

Enter the sequence to be folded in the box below. All non-alphabet characters will be removed. FASTA format may be used.

```
GGCCCCAUGCGAAGUUGU
UAUCGCGCCUCCUGUCAG
GAGGAGUACACGGUUCGAG
UCCGUUGGGUUCGCA
```

Enter **constraint information** in the box at the right. (optional) You may:

1. force bases $i+1, \dots, i+k-1$ to be double stranded by entering:
 $E _ i _ k$ on 1 line in the constraint box.
2. force consecutive base pairs $i, j+1, j-1, \dots, i+k-1, j-k+1$ by entering:
 $E _ i _ j _ k$ on 1 line in the constraint box.
3. force bases $i+1, \dots, i+k-1$ to be single stranded by entering:
 $P _ i _ k$ on 1 line in the constraint box.
4. prohibit the consecutive base pairs $i, j+1, j-1, \dots, i+k-1, j-k+1$ by entering:
 $P _ i _ j _ k$ on 1 line in the constraint box.
5. prohibit bases i or j from pairing with bases k to l by entering:
 $P _ i _ j _ k _ l$ on 1 line in the constraint box.

The RNA sequence is

Folding temperature is fixed at 37°.

Ionic conditions: 1M NaCl, no divalent ions.

Enter the **percent suboptimality** number:

(a)

2. force consecutive base pairs $i, j+1, j-1, \dots, i+k-1, j-k+1$ by entering:
 $E _ i _ j _ k$ on 1 line in the constraint box.

3. force bases $i+1, \dots, i+k-1$ to be single stranded by entering:
 $P _ i _ k$ on 1 line in the constraint box.

4. prohibit the consecutive base pairs $i, j+1, j-1, \dots, i+k-1, j-k+1$ by entering:
 $P _ i _ j _ k$ on 1 line in the constraint box.

5. prohibit bases i or j from pairing with bases k to l by entering:
 $P _ i _ j _ k _ l$ on 1 line in the constraint box.

The RNA sequence is

Folding temperature is fixed at 37°.

Ionic conditions: 1M NaCl, no divalent ions.

Enter the **percent suboptimality** number:

Enter an **upper bound** on the number of computed foldings:

Enter the **window** parameter if you wish:

Enter the **maximum interior/bulge loop size**

Enter the **maximum asymmetry of an interior/bulge loop**

Enter the **maximum distance between paired bases** if you wish:

Your job can be processed while you wait (the default) or can be submitted for batch processing by pressing the button below. In this case, you will be notified at a later time that the job is finished. If you select a **batch job**, please make sure your E-mail address is correct in the window below.

Select: job for:

Choose **image width** for png & jpg files: Small: Regular: Medium: Large: XLarge: Huge:

Choose **structure format**: Automatic: Bases: Outline:

Grid lines in **energy dot plot**: On: Off:

Choose **structure draw mode**:

Choose **exterior loop type**:

Choose **base numbering frequency**:

Choose **sequence numbering offset**:

Choose **regularization angle in degrees**: (Not used if 0.)

Choose **structure rotation angle**:

Choose **structure annotation**: None: p-num: ss-count: high-light:

Enter high-light regions(s):

Current limits: 800 bases for an immediate job, 9000 for batch.

If you wish to make comments, please select an appropriate [forum](#).

(b)

Figure 6.6 The input form for the version 3.1 Mfold server. (a) The top and (b) the bottom of the form. Default parameters are shown with the exceptions as noted in the text. Note that there is a separate server for secondary structure prediction of DNA, using DNA folding free energies (SantaLucia 1998). This is available by following the link to the DNA Mfold server (see Internet Resources).

value based on the length of the sequence that is shown by following the link at *Window*. For example, the tRNA used here is 77 nucleotides long and will have a default window of 2. A smaller window allows more suboptimal structures and a larger window requires more differences between the predicted structures. The smallest window size allowed is zero. The maximum number of unpaired nucleotides in bulge or internal loops is limited to 30 by default. The maximum asymmetry in internal loops, the difference in length in unpaired nucleotides on each strand, is also 30 by default. The maximum distance allowed between paired nucleotides is set to *no limit*. These values can be modified by advanced users.

The remaining options control the server output. Currently, sequences containing 800 or fewer nucleotides can be folded in a relatively short time and are treated as an immediate job. Longer sequences must be folded as a batch job, requiring that the default option be changed from *An immediate* to *A batch* job. Batch jobs also require that the user enter an e-mail address for receiving notification that the calculation is complete. The tRNA in this example is short, so the default of *An immediate* job will be used. The remaining options control the way the server generates output. Each of these options has a link to a web page that describes each parameter. *Fold RNA* is clicked to start the calculation.

Figure 6.7 shows the Mfold server output form for the secondary structure prediction of the RD1140 tRNA. Results are available on the server for 24 hours after computation. The first window displays the sequence with numbered nucleotide positions. A diagram of each predicted secondary structure is available in a variety of formats. For this example, only a single structure is predicted using the default parameters for suboptimal secondary structure prediction. The commonly used formats, available by links adjacent to *Structure 1*, are *PostScript*, which is a publication-quality output format shown in Figure 6.8a; *PNG* and *JPG*, which are image formats that allow user interaction; and *RNAviz CT* and *XRNA ss* formats, which are export formats for secondary structure drawing tools, explained below.

The energy dot plot is available by links to a *Text* formatted, *Postscript* formatted, *PNG* formatted, or *JPG* formatted file. In the dot plot, each dot represents a base pair between nucleotides indicated on the x- and y-axes and the dot's color indicates the lowest energy for a structure that

The screenshot shows the Mfold Web Server interface. The header includes 'THE RNA INSTITUTE COLLEGE OF ARTS AND SCIENCES UNIVERSITY AT ALBANY' and 'The mfold Web Server'. The main content area displays the following information:

- RD1140** is the 2948475th nucleic acid sequence folded on the RNA Institute mfold server.
- Computed for **urmc-nat18.urmc.rochester.edu**
- Folding RD1140 at 37° C. (3.5)**
- Linear RNA folding at 5%, window = 2, max folds = 50
- 12 A's, 23 C's, 26 G's, 16 U's and 0 N's.

The sequence is displayed as follows:

```

10      20      30      40      50
GGCCCCAUG CGAA GUUGGU UAUCGCCU CCCUGUCA G GAGGA GAUCA
   60      70      80
CGGUUCGAG UCCCGUGGG GUCCCA
  
```

The page also includes a sidebar with navigation links (Applications, View Folding Results, Documentation, Software, About, Contact) and an 'Output' section with links to file formats and a description of the energy dot plot.

Figure 6.7 The output page for the Mfold server. Please refer to the text for a detailed description of this page.

contains that pair. The energy dot plot is divided into two triangles. The upper triangle is the energy plot including suboptimal pairs and the lower triangle is the location of base pairs in the predicted minimum free energy structure. The text format is suitable for subsequent analysis using custom scripts. *Postscript* is a publication-quality output and is shown in Figure 6.8b. *PNG* and *JPG* formats both link to interactive pages that allow the user to zoom to regions, change the energy increment and number of colors, and click on individual base pairs to determine the exact energy. The energy dot plot in Figure 6.8b shows that there are alternative base pairs contained in structures with free energies between -29.8 and -30.0 kcal mol $^{-1}$, a separation of less than 1 kcal mol $^{-1}$ from the lowest free energy structure (-30.6 kcal mol $^{-1}$). Therefore, these are base pairs that should be considered as possible alternatives to those in the lowest free energy structure.

An RNAML formatted output file is available for exchanging information with other RNAML-compliant programs. This is an XML file format that promises to eventually allow seamless information exchange between RNA analysis programs (Waugh et al. 2002).

Using the RNAstructure Web Server

RNAstructure is a software package for predicting RNA secondary structure (Reuter and Mathews 2010; Bellaousov et al. 2013). In addition to implementations of the algorithms

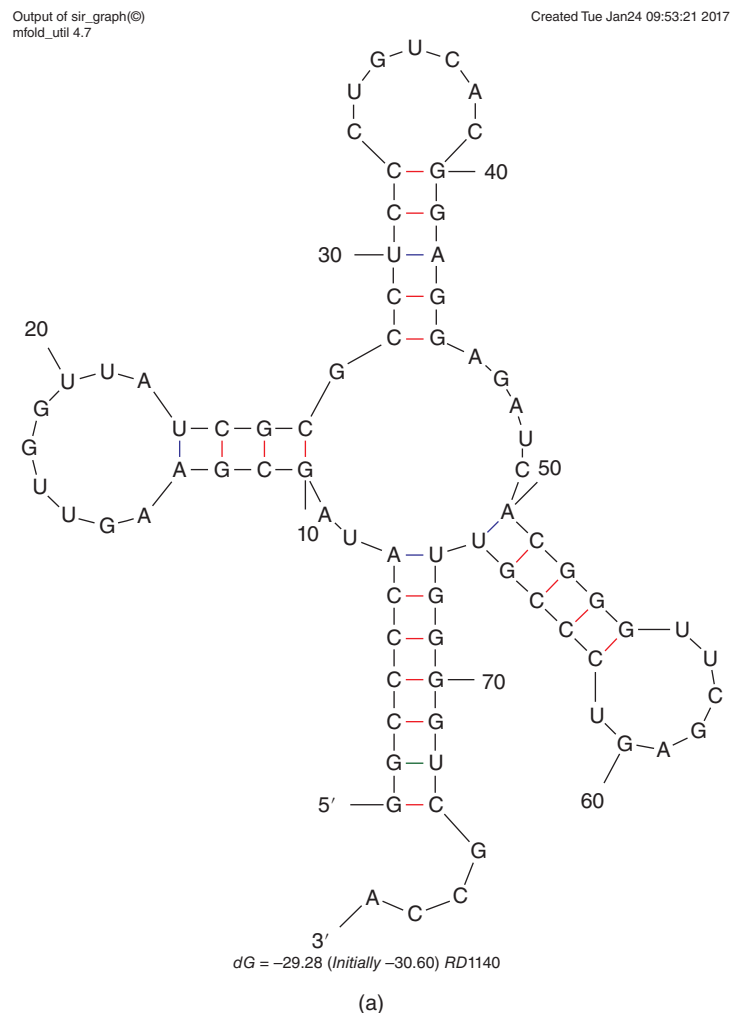
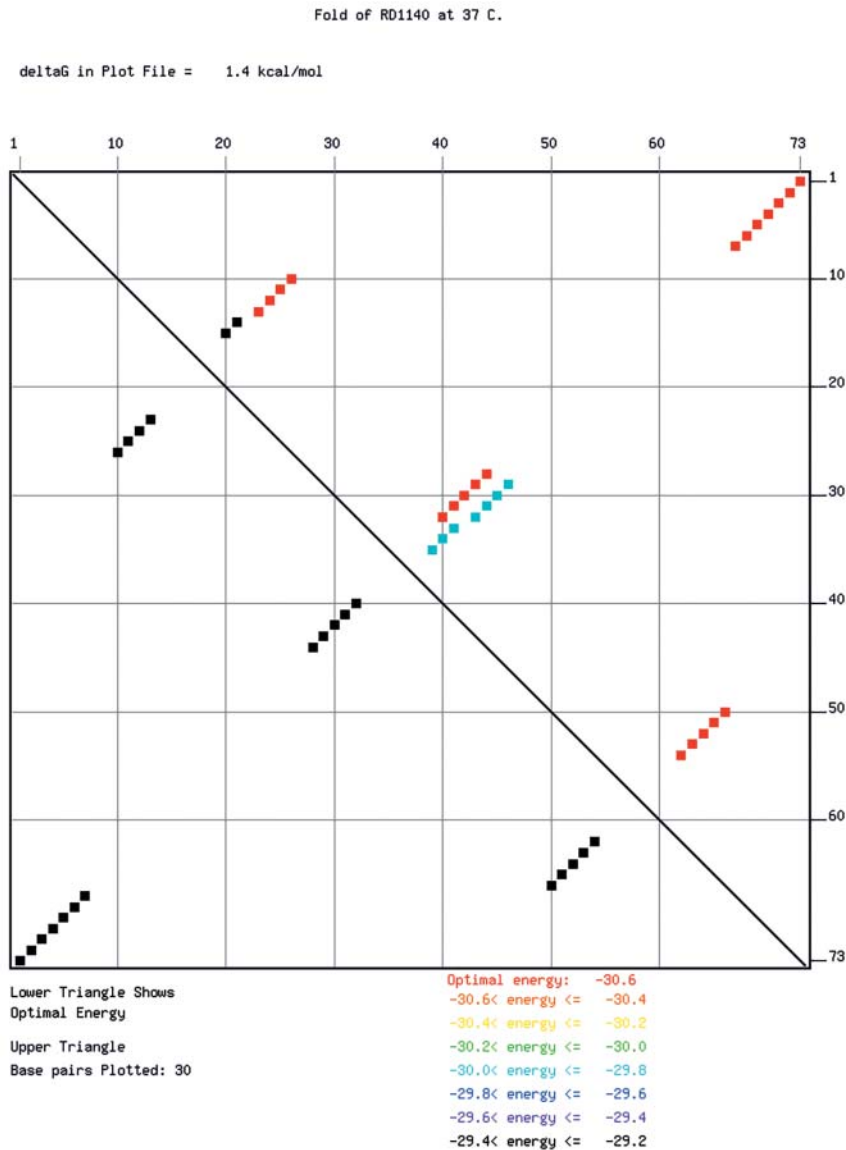


Figure 6.8 Sample output from the Mfold web server, version 3.1. (a) The secondary structure predicted for the tRNA, RD1140 (Sprinzl et al. 1998). (b) The energy dot plot (as described in the text).

Output of boxplot_ng (C)
mFold_util 4.7

Created Tue Jan 24 09:53:21 2017



(b)

Figure 6.8 (Continued)

for structure prediction and the partition function, RNAstructure includes tools for multiple-sequence prediction, identifying accessible regions in an RNA structure, and predicting the structure for two RNA molecules to hybridize. RNAstructure can be used through a web server or downloaded and used through a graphical user interface for Windows, Mac OS X, and Linux, or via the command line. This tutorial explains how to use the web server interface; detailed instructions for predicting secondary structures using the downloadable program are available in the online help files and elsewhere (Mathews et al. 2016; Xu and Mathews 2016).

The homepage for the RNAstructure web server offers options to predict a secondary structure from a single sequence, to predict a conserved structure with multiple homologous sequences, to predict the structure of two interacting sequences, or to run a specific algorithm

from the RNAstructure package. To predict a structure for a single sequence, choose *Predict a Secondary Structure*. This leads to an input form for the web server that predicts RNA structures using a selection of methods (Figure 6.9). Like Mfold, the minimum free energy structure for the input RNA sequence and a selection of suboptimal structures are computed. Additionally, base-pairing probabilities are calculated using the partition function and used to annotate the predicted RNA secondary structures, giving the user feedback about what parts of the predicted structure are most likely to be correct. Highly probable base pairs (i.e. higher than 0.9 pairing probability) are more likely to be correctly predicted than low probability (i.e. a pairing probability less than 0.5) (Mathews 2004). RNA secondary structures

Select Sequence Input:

Select Sequence File:

No file selected.

- OR -

Enter your sequence title and content below (replaces upload if present).
Please enter nucleotides only, no headers or comments in content.
Valid nucleotides are A, C, G, T, U, and X (unknown nucleotide). Lower case nucleotides are forced single stranded in calculation.

Sequence Title:

Sequence: (Click [here](#) to add an example sequence to the box.)

Select Nucleic Acid Type:

DNA:

RNA:

Select Predict a Secondary Structure Options:

Default Data

If a default value is left blank, the value is treated as if it was not changed at all.

Temperature (K):

Maximum Loop Size:

Maximum % Energy Difference (MFE, MEA):

Maximum Number of Structures (MFE, MEA):

Window Size (MFE, MEA):

(a)

Figure 6.9 RNAstructure web server input form. (a) The top and (b) the bottom of the form. Please refer to the text for a detailed description of this page.

<p>Gamma (MEA): <input type="text" value="1"/></p> <p>Iterations (Pseudoknot Prediction): <input type="text" value="1"/></p> <p>Minimum Helix Length (Pseudoknot Prediction): <input type="text" value="3"/></p>
<p>Optional Data</p> <p>Select Folding Constraints File: <input type="button" value="Browse..."/> No file selected.</p> <p><i>Note that in order to use SHAPE intercept and/or SHAPE slope, a SHAPE constraints file must be specified. If a value is given for intercept or slope without an accompanying constraints file, the intercept or slope value is not used in calculation.</i></p> <p>Select SHAPE Constraints File: <input type="button" value="Browse..."/> No file selected.</p> <p>SHAPE Intercept: <input type="text" value="-0.6"/></p> <p>SHAPE Slope: <input type="text" value="1.8"/></p> <p><i>Specify the minimum and maximum probabilities to show on the dot plot. If a specified probability is out of range, it is ignored.</i></p> <p>Minimum: <input type="text"/></p> <p>Maximum: <input type="text" value="2"/></p>
<p>Designate Address to Email Results:</p> <p>Address to Email Results: <input type="text"/></p> <p style="text-align: center;"><input type="button" value="Submit Query"/></p>

(b)

Figure 6.9 (Continued)

are also generated using the maximum expected accuracy method (Lu et al. 2009), which generates structures directly from the base-pairing probabilities calculated using the partition function, and can be more accurate than the minimum free energy structures. Additionally, secondary structures that may contain pseudoknots are generated using the ProbKnot method (Bellaousov and Mathews 2010).

Returning to the input form for the web server (Figure 6.9), the user can either upload a sequence file or type a title and sequence into the input box. There is an option above this box to insert an example sequence. The RNAstructure input form offers the same options as Mfold, such as number of suboptimal structures to generate, maximum internal loop size, and

others. Additionally, the user can select a temperature for folding; select a gamma parameter, which can be changed to increase or decrease the number of predicted base pairs by the maximum expected accuracy method; and the number of iterations and helix length used by the ShapeKnots method. Default values are provided that are expected to work well in most cases.

The form also contains an option to upload files that constrain or restrain the calculation with results from experiments. The user can either use hard constraints to forbid certain pairs or provide a file with scores from a SHAPE probing experiment that will be converted to pseudo-free energy changes to restrain structure prediction. Each of these files must be specified in a specific format, described in the *file formats* link.

An example output, showing the results for the example sequence, is shown in Figure 6.10. The structure is predicted using all free energy minimization, maximum expected accuracy prediction, and ProbKnot. Additionally, predicted secondary structures are color annotated with probabilities, as calculated with the partition function. Base-paired nucleotides are annotated with base-pairing probability and unpaired nucleotides are annotated with the probability of being unpaired.

Practical Introduction to Multiple Sequence Methods

Using the RNAstructure Web Server to Predict a Common Structure for Multiple Sequences

The RNAstructure web server also provides an interface to predict a secondary structure using multiple sequences, when sequence homologs are available. Homologs are most often sequences from multiple species that serve the same function (see Chapter 3 for definitions). They can be found in genomes using synteny or they can be found using traditional genetic or biochemical methods. The multiple sequence interface can be accessed by selecting *Predict a Secondary Structure Common to Two Sequences* or *Predict A Secondary Structure Common to Three or More Sequences* from the web server main page. Selecting the option for three or more sequences leads to the input form in Figure 6.11, for structure prediction using Multilign and TurboFold. To provide a sequence, the user can upload a file in the FASTA format or copy FASTA-formatted data into the box titled *Sequences*. There is an option above this box to enter example data.

Multilign uses Dynalign to calculate common structures using pairs of sequences (Xu and Mathews 2011). Multilign can predict suboptimal structures, and the input form contains the energy difference, maximum number of structures, and structure window size options that are by now familiar because they serve the same roles for single-sequence structure prediction (above). In addition to controlling suboptimal structures, suboptimal sequence alignments are also considered, using an alignment window size parameter. The minimum alignment window will generate suboptimal alignments with small variations. By setting this window size to larger values, the suboptimal alignments are required to show larger changes. The gap penalty parameter is used by Dynalign to penalize gaps inserted in sequence alignments. The penalty parameter is in units of kcal mol^{-1} . Two additional parameters, *iterations* and *maxdschange*, adjust the way information from Dynalign calculations propagates in Multilign. The default parameters should work for most calculations, and are available for change by experienced users (Xu and Mathews 2011).

For TurboFold, the user can specify a number of options. Because TurboFold produces a matrix of pair probabilities, a user can choose how the pair probabilities will be used to predict a structure: *Maximum Expected Accuracy* (Lu et al. 2009), *Pseudoknots* (Bellaousov and Mathews 2010; Seetin and Mathews 2012b), or *Threshold*, that uses a simple cut-off where the structure will be composed of all the pairs that exceed a user-specified threshold in base-pairing probability (Mathews 2004). The default is maximum expected accuracy, which does not predict pseudoknots. If a pseudoknot is expected or if the user would like

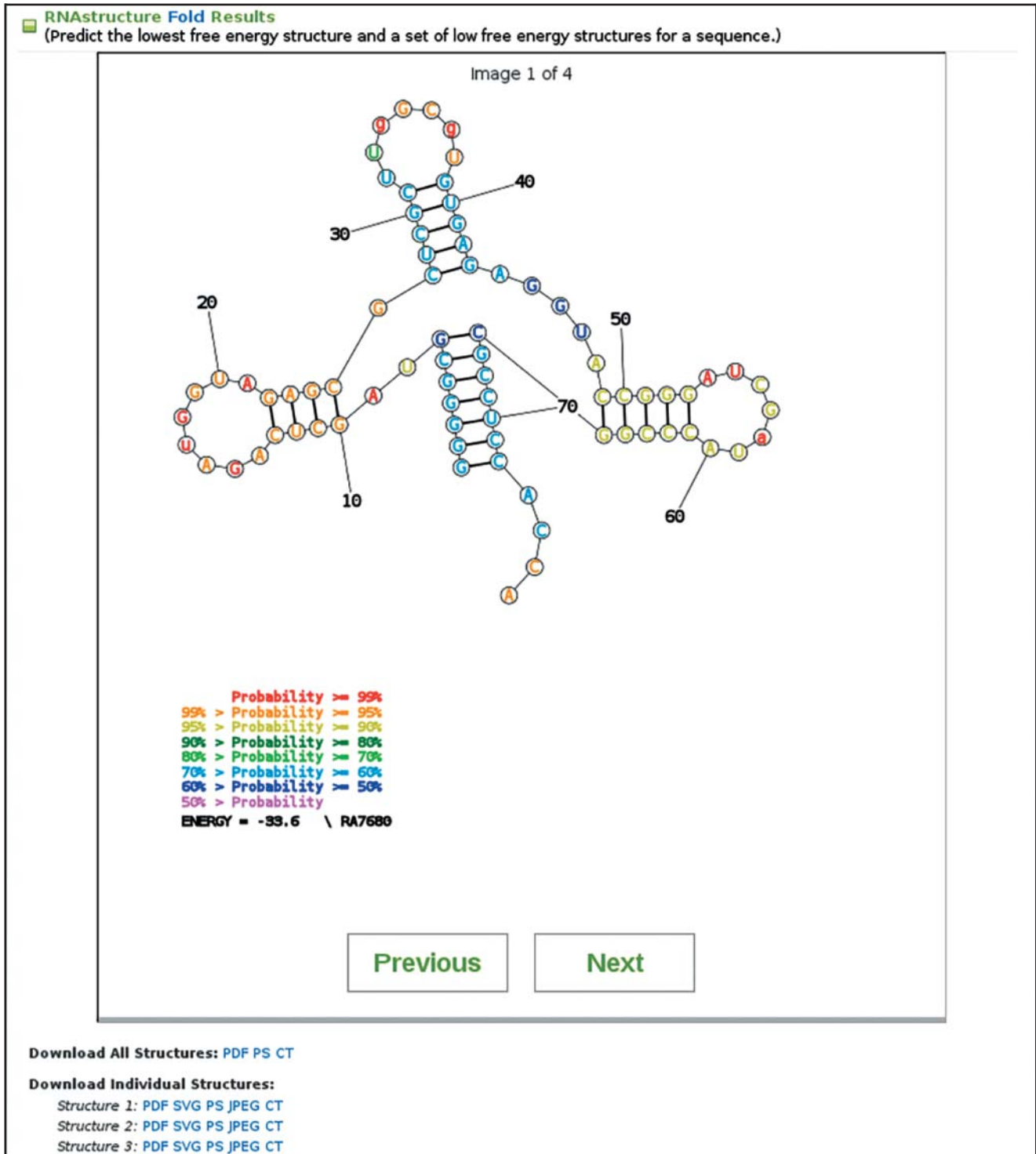


Figure 6.10 Sample output from the RNAstructure web server showing the predicted minimum free energy secondary structure for the tRNA RD1140 (Sprinzl et al. 1998). The predicted pairs are color annotated with their pairing probability, and the unpaired nucleotides are annotated with the probability that they are unpaired, as calculated using a partition function.

Select TurboFold Options:

Folding Mode

Maximum Expected Accuracy:

Pseudoknots:

Threshold:

Default Data
If a default value is left blank, the value is treated as if it was not changed at all.

General Options

TurboFold Gamma:

TurboFold Iterations:

Maximum Expected Accuracy Mode Options

Maximum % Energy Difference:

Maximum Number of Structures:

Window Size:

Maximum Expected Accuracy Gamma:

Pseudoknots Mode Options

Pseudoknot Iterations:

Minimum Helix Length:

Optional Data

Threshold Mode Options

Threshold:

Designate Address to Email Results:

Address to Email Results:

Submit Query

(a)

Figure 6.11 Input form for the RNAstructure web server for multiple-sequence predictions. (a) The top and (b) the bottom of the form. Please refer to the text for a detailed description of this page.

Welcome to the Predict a Secondary Structure Common to Three or More Sequences Web Server

The **Predict a Secondary Structure Common to Three or More Sequences** server takes **three or more sequences** and folds them into their common **lowest free energy conformations**. The Predict a Secondary Structure Common to Three or More Sequences server combines the capabilities of **Multalign** and **TurboFold** to create distinct sets of possible structures for multiple sequences. Note that **sequences must be entered as a group** in this server. Sequences can be entered either as a FASTA file with multiple sequences, or manually in the sequence box. Note also that the server **assumes the sequences are RNA**.

If you plan to leave this page while waiting for calculations to complete, please **make sure you have entered a valid email address** below. Leaving this page without entering a valid email address will render results inaccessible.

If you need specific **help using the Predict a Secondary Structure Common to Three or More Sequences server**, please click [here](#).

Select Sequences Input:

Select Sequences File:

No file selected.

- OR -

Enter your sequence content below (replaces upload if present).
 Sequence content should be entered in FASTA format, with a proper FASTA header on a line preceding each sequence.
 Please enter headers and nucleotides only, no comments in content.
 Valid nucleotides are A, C, G, T, U, and X (unknown nucleotide). Lower case nucleotides are forced single stranded in calculation.

Sequences: (Click [here](#) to add example sequences to the box.)

Select General Options:

Default Data
If a default value is left blank, the value is treated as if it was not changed at all.

Temperature (K):

Select Multalign Options:

Default Data
If a default value is left blank, the value is treated as if it was not changed at all.

Iterations:

Maxdschange:

Maximum % Energy Difference:

Maximum Number of Structures:

Structure Window Size:

Alignment Window Size:

(b)

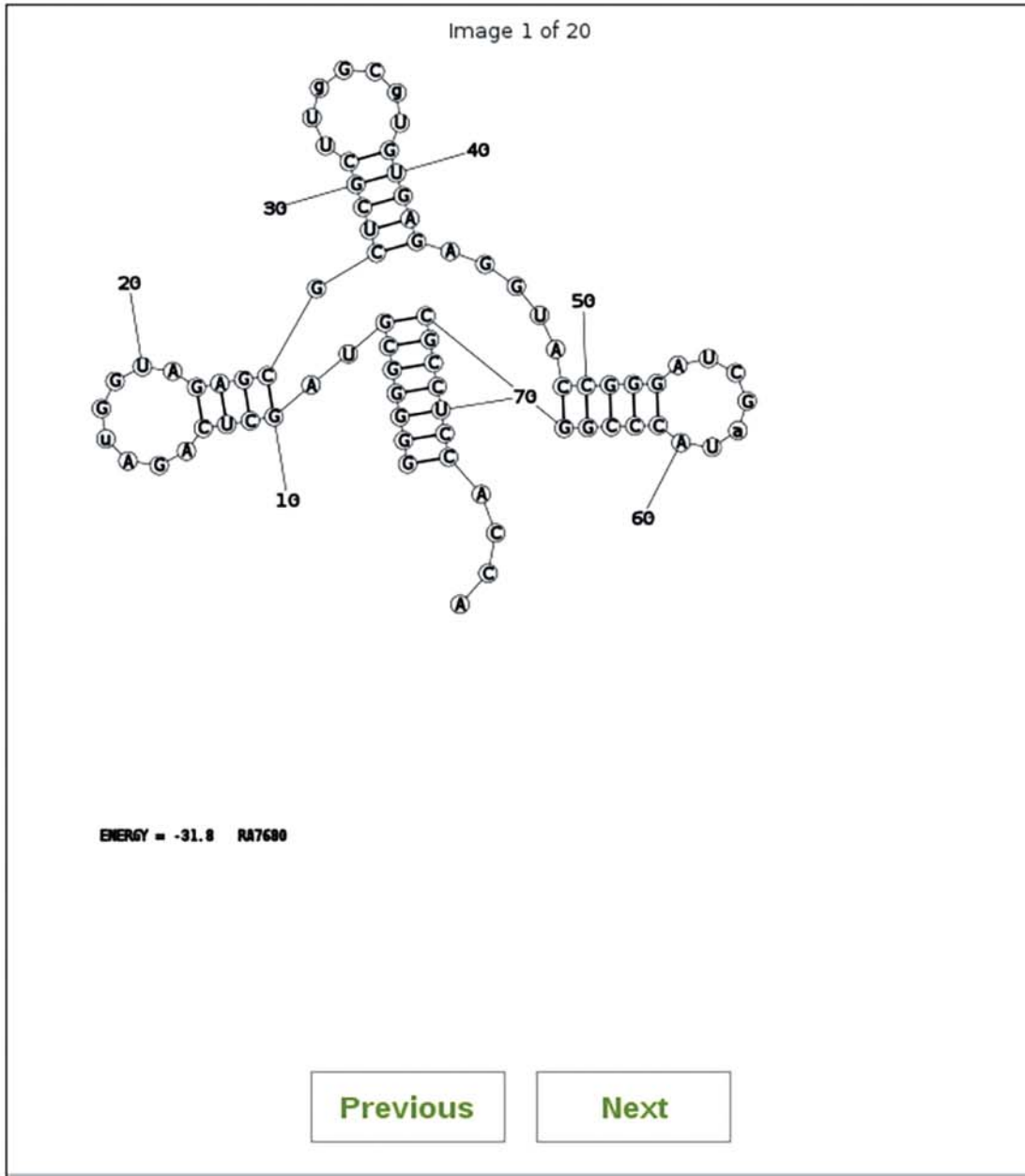
Figure 6.11 (Continued)

RNAstructure multalign Results

■ (Predict low free energy secondary structures common to three or more sequences using progressive iterations of Dynalign.)

[Click here to download the alignment file.](#)

Structures derived from sequence 1:



Download All Structures: [PDF](#) [PS](#) [CT](#)

Download Individual Structures:

Structure 1: [PDF](#) [SVG](#) [PS](#) [JPEG](#) [CT](#)

Structure 2: [PDF](#) [SVG](#) [PS](#) [JPEG](#) [CT](#)

Structure 3: [PDF](#) [SVG](#) [PS](#) [JPEG](#) [CT](#)

Structure 4: [PDF](#) [SVG](#) [PS](#) [JPEG](#) [CT](#)

Structure 5: [PDF](#) [SVG](#) [PS](#) [JPEG](#) [CT](#)

Structure 6: [PDF](#) [SVG](#) [PS](#) [JPEG](#) [CT](#)

Figure 6.12 Sample output from the RNAstructure web server for multiple-sequence predictions that can be accessed by clicking *Click here to add example sequences to the box* on the input form (Figure 6.11).

to know about possible pseudoknots, switching to Pseudoknot, which uses the ProbKnot method, would be a good choice. The next set of options controls the TurboFold procedure: the *TurboFold Gamma* option specifies the relative weight placed on intrinsic and extrinsic information when folding a single sequence, and the *TurboFold Iterations* option specifies how many steps of iterative refinement each set of pair probabilities will undergo.

Entering the example data and selecting *Submit Query* leads to the output shown in Figure 6.12, which displays a predicted structure by both Multalign and TurboFold for each input sequence. These two programs provide alternative hypotheses for the secondary structure; Multalign is likely to be more accurate when the sequences have little pairwise sequence identity (<50%) and TurboFold is likely to be more accurate with high pairwise sequence identities (>60%).

Other Computational Methods to Study RNA Structure

Another widely used RNA structure prediction package is the ViennaRNA package (Lorenz et al. 2011). ViennaRNA is available as a web server and as a set of locally run command line tools. The ViennaRNA package programs have a UNIX-style input, accepting data from standard input and printing results to standard output, making them easy to seamlessly integrate into a UNIX pipeline. In addition to implementations of the usual minimum free energy and partition function algorithms, ViennaRNA includes a suite of tools for predicting a common structure for multiple sequences, for drawing structures, for predicting duplex structures between two RNA chains, and for designing sequences that fold to a desired structure.

Sfold is an implementation of the partition function calculation that predicts secondary structures using a stochastic sampling procedure (Ding and Lawrence 1999, 2001, 2003). The sampling procedure guarantees that structures are sampled with true statistical weight. Sfold is available for use through a web interface. Sfold has been shown to accurately predict unpaired regions that correlate to regions accessible to antisense oligonucleotide targeting (Ding and Lawrence 2001). As the secondary structures are sampled statistically, the fraction of occurrences that a nucleotide is unpaired in a set of sampled structures is the predicted probability for being unpaired.

NUPACK (Zadeh et al. 2010) is a software suite (available through a web server and as a downloadable package) that solves the *inverse* folding problem, the opposite problem from RNA structure prediction. Instead of taking a sequence and predicting its structure, NUPACK takes a structure as input and attempts to find a sequence that will fold to that structure. Because the sequence space to be searched is enormous (4^N for a sequence of length N , so there are approximately 1.6×10^{60} possible sequences for a 100-nucleotide RNA), NUPACK hierarchically decomposes the structure into components. Sequences are designed for each component. The component sequences are then assembled, and, if any combination of sequences fails, the components are redesigned. Good candidate sequences are found by optimizing the ensemble defect, a quantity calculated from the partition function that estimates how many bases in the sequence are forming the desired structure (Zadeh et al. 2011).

Another important problem is to predict whether two RNA molecules will hybridize with one another, and the structure of the resulting duplex. Important applications of this capability are to predict targets of siRNA (Lu and Mathews 2007; Tafer et al. 2008), miRNA, or DNA oligonucleotides (Mathews et al. 1999b). The most accurate approaches for prediction of RNA–RNA interactions consider the balance between self-structure and intermolecular structure because the self-structures can prevent binding by a second sequence. Implementations of multi-sequence folding include RNAup (Muckstein et al. 2006) and RNAplex (Tafer and Hofacker 2008), which are part of the ViennaRNA package, and BiFold (Mathews et al. 1999b), DuplexFold (Mathews et al. 1999b), and AccessFold (DiChiacchio et al. 2016), which are components of RNAstructure. Web servers are available for RNAup (as part of the ViennaRNA web server), BiFold (as part of the RNAstructure web server), and DuplexFold

(as part of the RNAstructure web server). AccessFold and RNAplex are not available through web servers, but can be downloaded and run locally.

Comparison of Methods

No single program is yet available that can replace manual comparative sequence analysis, but additional sources of information can dramatically improve prediction accuracy. When experimental data such as SHAPE are available, single-sequence thermodynamic prediction can often provide accurate secondary structures, correctly predicting 90% or more of the known base pairs (Deigan et al. 2009; Cordero et al. 2012; Hajdin et al. 2013). If multiple homologs are available, multiple-sequence predictions of the conserved secondary structure are more accurate than single-sequence predictions (Asai and Hamada 2014; Havgaard and Gorodkin 2014). Dynalign/Multalign (Fu et al. 2014), FoldAlign/FoldAlignM (Havgaard et al. 2005), LocARNA (Will et al. 2007), RAF (Do et al. 2008), or TurboFold (Harmanci et al. 2011) can be helpful for simultaneously predicting secondary structure and aligning sequences that are too dissimilar to be aligned by primary sequence without using secondary structure (Mathews and Turner 2002). These programs have similar average accuracy. TurboFold is the fastest, but Dynalign, Foldalign, LocARNA, or RAF probably perform with higher accuracy when the sequence identity is low (average pairwise percent sequence identity less than 35%). It might be worth making predictions with more than one software tool to obtain multiple hypotheses for the conserved structure. For a set of homologs with high sequence identity (>85%), RNAalifold is another excellent tool for predicting the structure conserved by multiple homologs (Bernhart et al. 2008). It requires a multiple sequence alignment as input. The methods, like RNAalifold, for finding secondary structure in multiple sequence alignments are best used as screening tools to find common helices, which can be used to anchor portions of a sequence alignment when making manual revisions for further rounds of analysis.

Predicting RNA Tertiary Structure

Although there are many automated methods for accurate RNA secondary structure prediction, RNA tertiary structure prediction remains a more difficult problem. This is because the space of possible tertiary structures is much larger than the space of possible secondary structures, and there is no known algorithm that can search the conformational space as quickly or completely as can be done for secondary structure prediction.

A pioneering approach to tertiary structure prediction was implemented in the MC-SYM software (Major et al. 1991, 1993; Parisien and Major 2008). With an approach called fragment assembly, MC-SYM builds structural models by assembling nucleotides in conformations collected from known structures. Each possible model is stored until it is shown to contradict a constraint, based on experimental data, comparative analysis, or secondary structure prediction. The variations between all compatible models can suggest how accurately the model has been determined with the data used. An early demonstration of the utility of MC-SYM was the modeling of the hairpin ribozyme using data about secondary structure, hydroxyl radical footprinting, photoaffinity cross-linking, and disulfide cross-linking (Pinar et al. 1999). A subsequent crystal structure verified the existence of a predicted long-range GC pair, although a predicted base triple, involving an A at that pair, was not observed (Rupert and Ferré-D'Amaré 2001). More recently, the algorithm was used in concert with an extended secondary structure prediction method called MC-Fold, which can predict some tertiary interactions, to accurately predict structures for RNA molecules up to 100 nucleotides in length (Parisien and Major 2008).

Another RNA fragment assembly approach is based on the Rosetta framework (Cheng et al. 2015), which has been highly successful in protein structure prediction (Simons et al. 1997).

This is a fragment assembly approach used in concert with a knowledge-based force field that is used to sample native-like conformations.

Another class of de novo tertiary structure prediction methods is to use physics-based molecular dynamics (MD) simulations to predict structures. However, these MD simulations are far too slow to allow for structure prediction by themselves. A typical MD simulation might take weeks to run, even on specialized hardware, and simulate mere microseconds of folding. In nature, even simple RNA molecules take milliseconds or longer (Turner 2000) to fold, and complex structures can take seconds (Woodson 2000). In order to bridge this gap, two broad approaches can be used. One approach is to add constraints to the simulation based on computational structure prediction, low-resolution experimental data, and sequence comparison (Seetin and Mathews 2011; Weinreb et al. 2016). Another is to use coarse-graining; that is, replacing multiple atoms in the real molecule with a single “pseudo-atom,” therefore reducing the number of degrees of freedom available for sampling (Flores and Altman 2010; Krokhotin et al. 2015). This dramatically speeds up simulations, at a cost of detailed atomic accuracy in the resulting coordinates.

For RNA sequences for which there is already a high-resolution experimental structure for a closely related sequence, another prediction method, called homology modeling, can be used. Here, the structure for the homolog and its alignment with the new sequence are used to generate a structure for the new sequence, making the assumption that the new structure deviates from the old structure only in relatively minor details. Structure predictions generated via homology modeling can be highly accurate while remaining fast, because only a relatively small conformational space – the space of structures highly similar to the model of the homolog – needs to be sampled. Homology modeling for RNA is implemented in the ModeRNA program, for which a web server is available (Rother et al. 2011).

A recent multi-group effort, called RNA-PUZZLES, is an attempt to use a friendly competition to evaluate the progress of RNA tertiary structure prediction methods (Cruz et al. 2012; Miao et al. 2015). This is a blind RNA structure prediction contest modeled after the Critical Assessment of Structure Prediction (CASP) challenge, an analogous contest for protein structure (Moult et al. 2016). The RNA-PUZZLES project recruits structural biologists who solve novel RNA structures to share their coordinates. The sequence of the RNA molecule that has been solved is then shared with the computational modelers, who each provide their best guess as to the three-dimensional structure that the sequence will adopt. Finally, after the experimental structure is published, the computational models are compared with the experiment and assessed for accuracy.

The results of the first two rounds of RNA-PUZZLES revealed that modelers can accurately predict the overall topology of an RNA molecule from its sequence. However, structural details are often predicted incorrectly, especially loop regions, whose structures are determined by non-canonical contacts that are poorly predicted. Accurate modeling of loop regions will be an important step in the improvement of RNA tertiary structure prediction.

Summary

RNA secondary structure can be predicted by free energy minimization using dynamic programming with an average of 73% accuracy for a single sequence (Mathews et al. 2004). Several software packages and web servers, including Mfold, the Vienna package, and RNAstructure, are available to do this calculation (Hofacker 2003; Zuker 2003; Reuter and Mathews 2010). Calculation of pairing probabilities using a partition function can help in the identification of base pairs that are not well determined (Mathews 2004). The partition function can also be used to stochastically sample structures from the ensemble of possible structures, and this capability is used to predict structures by the Sfold program (Ding and Lawrence 2003).

Several methods are available to constrain secondary structure prediction using multiple sequences and multiple sequence alignment. These are divided among algorithms that are limited to an initial sequence alignment and those that are not limited to an initial alignment. Alifold and Pfold predict a secondary structure common to a set of aligned sequences (Knudsen and Hein 1999; Hofacker et al. 2002). Dynalign, FoldAlign, LocARNA, and TurboFold are capable of simultaneously predicting a common secondary structure and sequence alignment (Havgaard et al. 2005; Will et al. 2007; Harmanci et al. 2011; Fu et al. 2014). For long sequences or alignments of large numbers of homologs, TurboFold and LocARNA implement fast algorithms that provide good accuracy.

An important recent development has been the experimental methods used to probe RNA secondary structure, in a high-throughput fashion, inside living cells (Spitale et al. 2013; Ding et al. 2014; Rouskin et al. 2014; Talkish et al. 2014). These methods, used in concert with computational structure prediction methods (Deigan et al. 2009; Hajdin et al. 2013), will surely continue to yield new insights into RNA structure and function.

An important need in the field of RNA secondary structure prediction is improved methods to predict RNA–RNA interactions. Tools such as AccessFold (DiChiacchio et al. 2016) and RNAup (Muckstein et al. 2006) are an improvement over their predecessors, but still lack sufficient accuracy to solve many practical problems.

RNA tertiary structure prediction is rapidly improving, but still remains difficult. In particular, while the helical regions and the overall topology of a molecule can be correctly modeled, many atomic details remain inaccurate. The RNA-PUZZLES contest provides an ongoing measure of the rapid improvements in this field (Cruz et al. 2012; Miao et al. 2015).

Internet Resources

Mfold	unafold.rna.albany.edu/?q=mfold
ModeRNA	iimcb.genesilico.pl/modernaserver
Nearest Neighbor Database (NNDB)	rna.urmc.rochester.edu/NNDB
RNAstructure	rna.urmc.rochester.edu/RNAstructure.html
Sfold	sfold.wadsworth.org/cgi-bin/index.pl
ViennaRNA Package	rna.tbi.univie.ac.at
Wikipedia RNA Software Page	en.wikipedia.org/wiki/List_of_RNA_structure_prediction_software

Further Reading

- Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998). *Biological Sequence Analysis. Probabilistic Models of Proteins and Nucleic Acids*. New York, NY: Cambridge University Press This book is an excellent primer on probabilistic models for sequence analysis, including hidden Markov models and stochastic context-free grammars.
- Gorodkin, J. and Ruzzo, W.L. (eds.) (2014). *RNA Sequence, Structure, and Function: Computational and Bioinformatic Methods*. New York, NY: Humana Press.
- Turner, D.H. and Mathews, D.H. (2009). NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic Acids Res.* 38: D280–D282. This describes NNDB, which provides the latest nearest neighbor parameters and usage examples.

References

- Andronescu, M., Condon, A., Turner, D.H., and Mathews, D.H. (2014). The determination of RNA folding nearest neighbor parameters. *Methods Mol. Biol.* 1097: 45–70.

- Asai, K. and Hamada, M. (2014). RNA structural alignments, part II: non-Sankoff approaches for structural alignments. *Methods Mol. Biol.* 1097: 291–301.
- Bellaousov, S. and Mathews, D.H. (2010). ProbKnot: fast prediction of RNA secondary structure including pseudoknots. *RNA* 16: 1870–1880.
- Bellaousov, S., Reuter, J.S., Seetin, M.G., and Mathews, D.H. (2013). RNAstructure: web servers for RNA secondary structure prediction and analysis. *Nucleic Acids Res.* 41 (Web Server issue): W471–W474.
- Bernhart, S.H., Hofacker, I.L., Will, S. et al. (2008). RNAalifold: improved consensus structure prediction for RNA alignments. *BMC Bioinf.* 9: 474.
- Bittker, J., Phillips, K., and Liu, D. (2002). Recent advances in the *in vitro* evolution of nucleic acids. *Curr. Opin. Chem. Biol.* 6: 367–374.
- Brown, J.W. (1999). The ribonuclease P database. *Nucleic Acids Res.* 27: 314.
- Cannone, J.J., Subramanian, S., Schnare, M.N. 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 Bioinf.* 3 (2).
- Castanotto, D. and Rossi, J.J. (2009). The promises and pitfalls of RNA-interference-based therapeutics. *Nature* 457 (7228): 426–433.
- Cate, J.H., Gooding, A.R., Podell, E. et al. (1996). Crystal structure of a group I ribozyme domain: principles of RNA packing. *Science* 273: 1678–1685.
- Cheng, C.Y., Chou, F.C., and Das, R. (2015). Modeling complex RNA tertiary folds with Rosetta. *Methods Enzymol.* 553: 35–64.
- Condon, A., Davy, B., Rastegari, B. et al. (2004). Classifying RNA Pseudoknotted structures. *Theor. Comput. Sci.* 320: 35–50.
- Cordero, P., Kladwang, W., VanLang, C.C., and Das, R. (2012). Quantitative dimethyl sulfate mapping for automated RNA secondary structure inference. *Biochemistry* 51 (36): 7037–7039.
- Cruz, J.A., Blanchet, M.F., Boniecki, M. et al. (2012). RNA-Puzzles: a CASP-like evaluation of RNA three-dimensional structure prediction. *RNA*. 18 (4): 610–625.
- Cullen, B.R. (2002). RNA interference: antiviral defense and genetic tool. *Nat. Immunol.* 3: 597–599.
- Deigan, K.E., Li, T.W., Mathews, D.H., and Weeks, K.M. (2009). Accurate SHAPE-directed RNA structure determination. *Proc. Natl. Acad. Sci. USA.* 106 (1): 97–102.
- Dias, N. and Stein, C.A. (2002). Antisense oligonucleotides: basic concepts and mechanisms. *Mol. Cancer Ther.* 1: 347–355.
- DiChiacchio, L., Sloma, M.F., and Mathews, D.H. (2016). AccessFold: predicting RNA-RNA interactions with consideration for competing self-structure. *Bioinformatics* 32: 1033–1039.
- Dimitrov, R.A. and Zuker, M. (2004). Prediction of hybridization and melting for double-stranded nucleic acids. *Biophys. J.* 87 (1): 215–226.
- Ding, Y. and Lawrence, C.E. (1999). A Bayesian statistical algorithm for RNA secondary structure prediction. *Comput. Chem.* 23: 387–400.
- Ding, Y. and Lawrence, C. (2001). Statistical prediction of single-stranded regions in RNA secondary structure and application to predicting effective antisense target sites and beyond. *Nucleic Acids Res.* 29: 1034–1046.
- Ding, Y. and Lawrence, C.E. (2003). A statistical sampling algorithm for RNA secondary structure prediction. *Nucleic Acids Res.* 31 (24): 7280–7301.
- Ding, Y., Tang, Y., Kwok, C.K. et al. (2014). In vivo genome-wide profiling of RNA secondary structure reveals novel regulatory features. *Nature*. 505 (7485): 696–700.
- Dirks, R.M. and Pierce, N.A. (2003). A partition function algorithm for nucleic acid secondary structure including pseudoknots. *J. Comput. Chem.* 24: 1664–1677.
- Disney, M.D., Winkelsas, A.M., Velagapudi, S.P. et al. (2016). Inforna 2.0: a platform for the sequence-based design of small molecules targeting structured RNAs. *ACS Chem. Biol.* 11 (6): 1720–1728.
- Do, C.B., Foo, C.S., and Batzoglou, S. (2008). A max-margin model for efficient simultaneous alignment and folding of RNA sequences. *Bioinformatics.* 24 (13): i68–i76.

- Doudna, J.A. and Cech, T.R. (2002). The chemical repertoire of natural ribozymes. *Nature*. 418: 222–228.
- Flores, S.C. and Altman, R.B. (2010). Turning limited experimental information into 3D models of RNA. *RNA*. 16 (9): 1769–1778.
- Fu, Y., Sharma, G., and Mathews, D.H. (2014). Dynalign II: common secondary structure prediction for RNA homologs with domain insertions. *Nucleic Acids Res.* 42 (22): 13939–13948.
- Gutell, R.R., Lee, J.C., and Cannone, J.J. (2002). The accuracy of ribosomal RNA comparative structure models. *Curr. Opin. Struct. Biol.* 12: 301–310.
- Hajdin, C.E., Bellaousov, S., Huggins, W. et al. (2013). Accurate SHAPE-directed RNA secondary structure modeling, including pseudoknots. *Proc. Natl. Acad. Sci. USA*. 110 (14): 5498–5503.
- Halvorsen, M., Martin, J.S., Broadaway, S., and Laederach, A. (2010). Disease-associated mutations that alter the RNA structural ensemble. *PLoS Genet.* 6 (8): e1001074.
- Hansen, J.L., Schmeing, T.M., Moore, P.B., and Steitz, T.A. (2002). Structural insights into peptide bond formation. *Proc. Natl. Acad. Sci. USA*. 99: 11670–11675.
- Harmanci, A.O., Sharma, G., and Mathews, D.H. (2008). PARTS: probabilistic alignment for RNA joinT secondary structure prediction. *Nucleic Acids Res.* 36: 2406–2417.
- Harmanci, A.O., Sharma, G., and Mathews, D.H. (2009). Stochastic sampling of the RNA structural alignment space. *Nucleic Acids Res.* 37: 4063–4075.
- Harmanci, A.O., Sharma, G., and Mathews, D.H. (2011). TurboFold: iterative probabilistic estimation of secondary structures for multiple RNA sequences. *BMC Bioinf.* 12 (1): 108.
- Havgaard, J.H. and Gorodkin, J. (2014). RNA structural alignments, part I: Sankoff-based approaches for structural alignments. *Methods Mol. Biol.* 1097: 275–290.
- Havgaard, J.H., Lyngso, R.B., Stormo, G.D., and Gorodkin, J. (2005). Pairwise local structural alignment of RNA sequences with sequence similarity less than 40%. *Bioinformatics*. 21 (9): 1815–1824.
- Hofacker, I.L. (2003). Vienna RNA secondary structure server. *Nucleic Acids Res.* 31: 3429–3431.
- Hofacker, I.L., Fekete, M., and Stadler, P.F. (2002). Secondary structure prediction for aligned RNA sequences. *J. Mol. Biol.* 319: 1059–1066.
- Jaeger, J.A., Turner, D.H., and Zuker, M. (1989). Improved predictions of secondary structures for RNA. *Proc. Natl. Acad. Sci. USA* 86: 7706–7710.
- Juan, V. and Wilson, C. (1999). RNA secondary structure prediction based on free energy and phylogenetic analysis. *J. Mol. Biol.* 289: 935–947.
- Knapp, G. (1989). Enzymatic approaches to probing RNA secondary and tertiary structure. *Methods Enzymol.* 180: 192–212.
- Knudsen, B. and Hein, J. (1999). RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. *Bioinformatics*. 15: 446–454.
- Krokhotin, A., Houlihan, K., and Dokholyan, N.V. (2015). iFoldRNA v2: folding RNA with constraints. *Bioinformatics*. 31 (17): 2891–2893.
- Lagos-Quintana, M., Rauhut, R., Lendeckel, W., and Tuschl, T. (2001). Identification of novel genes coding for small expressed RNAs. *Science*. 294: 853–858.
- Larsen, N., Samuelsson, T., and Zwieb, C. (1998). The signal recognition particle database (SRPDB). *Nucleic Acids Res.* 26: 177–178.
- Lathe, W.C. III. and Eickbush, T.H. (1997). A single lineage of R2 retrotransposable elements is an active, evolutionarily stable component of the *Drosophila* rDNA locus. *Mol. Biol. Evol.* 14: 1232–1241.
- Lau, N.C., Lim, L.P., Weinstein, E.G., and Bartel, D.P. (2001). An abundant class of tiny RNAs with probable regulatory roles in *Caenorhabditis elegans*. *Science*. 294: 858–862.
- Layton, D.M. and Bundschuh, R. (2005). A statistical analysis of RNA folding algorithms through thermodynamic parameter perturbation. *Nucleic Acids Res.* 33 (2): 519–524.
- Liu, B., Mathews, D.H., and Turner, D.H. (2010). RNA pseudoknots: folding and finding. *F1000 Biol. Rep.* 2: 8.
- Lorenz, R., Bernhart, S.H., Honer Zu Siederdisen, C. et al. (2011). ViennaRNA Package 2.0. *Algorithms Mol. Biol.* 6: 26.

- Lu, Z.J. and Mathews, D.H. (2007). Efficient siRNA selection using hybridization thermodynamics. *Nucleic Acids Res.* 36: 640–647.
- Lu, Z.J., Gloor, J.W., and Mathews, D.H. (2009). Improved RNA secondary structure prediction by maximizing expected pair accuracy. *RNA*. 15: 1805–1813.
- Lück, R., Steger, G., and Riesner, D. (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: 813–826.
- Lück, R., Gräf, S., and Steger, G. (1999). ConStruct: a tool for thermodynamic controlled prediction of conserved secondary structure. *Nucleic Acids Res.* 27: 4208–4217.
- Major, F., Turcotte, M., Gautheret, D. et al. (1991). The combination of symbolic and numerical computation for three-dimensional modeling of RNA. *Science*. 253: 1255–1260.
- Major, F., Gautheret, D., and Cedergren, R. (1993). Reproducing the three-dimensional structure of a tRNA molecule from structural constraints. *Proc. Natl. Acad. Sci. USA*. 90: 9408–9412.
- Mathews, D.H. (2004). Using an RNA secondary structure partition function to determine confidence in base pairs predicted by free energy minimization. *RNA*. 10: 1178–1190.
- Mathews, D.H. and Turner, D.H. (2002). Dynalign: an algorithm for finding the secondary structure common to two RNA sequences. *J. Mol. Biol.* 317: 191–203.
- Mathews, D.H., Banerjee, A.R., Luan, D.D. et al. (1997). Secondary structure model of the RNA recognized by the reverse transcriptase from the R2 retrotransposable element. *RNA*. 3: 1–16.
- Mathews, D.H., Sabina, J., Zuker, M., and Turner, D.H. (1999a). Expanded sequence dependence of thermodynamic parameters provides improved prediction of RNA secondary structure. *J. Mol. Biol.* 288: 911–940.
- Mathews, D.H., Burkard, M.E., Freier, S.M. et al. (1999b). Predicting oligonucleotide affinity to nucleic acid targets. *RNA*. 5: 1458–1469.
- Mathews, D.H., Disney, M.D., Childs, J.L. et al. (2004). Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proc. Natl. Acad. Sci. USA*. 101: 7287–7292.
- Mathews, D.H., Turner, D.H., and Watson, R.M. (2016). RNA secondary structure prediction. *Curr. Protoc. Nucleic Acid Chem.* 67: 11.12.1–11.12.19.
- Miao, Z., Adamiak, R.W., Blanchet, M.F. et al. (2015). RNA-puzzles round II: assessment of RNA structure prediction programs applied to three large RNA structures. *RNA*. 21 (6): 1066–1084.
- Michel, F., Umeshima, K., and Ozeki, H. (1989). Comparative and functional anatomy of group II catalytic introns – a review. *Gene*. 82: 5–30.
- Moult, J., Fidelis, K., Kryshtafovych, A. et al. (2016). Critical assessment of methods of protein structure prediction: Progress and new directions in round XI. *Proteins*. 84 (Suppl 1): 4–14.
- Muckstein, U., Tafer, H., Hackermüller, J. et al. (2006). Thermodynamics of RNA-RNA binding. *Bioinformatics*. 22 (10): 1177–1182.
- Nissen, P., Hansen, J., Ban, N. et al. (2000). The structural basis of ribosomal activity in peptide bond synthesis. *Science*. 289: 920–930.
- Nussinov, R. and Jacobson, A.B. (1980). Fast algorithm for predicting the secondary structure of single-stranded RNA. *Proc. Natl. Acad. Sci. USA*. 77: 6309–6313.
- Pace, N.R., Thomas, B.C., and Woese, C.R. (1999). Probing RNA structure, function, and history by comparative analysis. In: *The RNA World*, 2e (eds. R.F. Gesteland, T.R. Cech and J.F. Atkins), 113–141. Cold Spring Harbor, NY: Cold Spring Harbor Laboratory Press.
- Parisien, M. and Major, F. (2008). The MC-fold and MC-Sym pipeline infers RNA structure from sequence data. *Nature*. 452 (7183): 51–55.
- Pinard, R., Lambert, D., Walter, N.G. et al. (1999). Structural basis for the guanosine requirement of the hairpin ribozyme. *Biochemistry*. 38: 16035–16039.
- Reeder, J. and Giegerich, R. (2004). Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics. *BMC Bioinf.* 5: 104.
- Reuter, J.S. and Mathews, D.H. (2010). RNAstructure: software for RNA secondary structure prediction and analysis. *BMC Bioinf.* 11: 129.

- Rivas, E. and Eddy, S.R. (1999). A dynamic programming algorithm for RNA structure prediction including pseudoknots. *J. Mol. Biol.* 285: 2053–2068.
- Rother, M., Rother, K., Puton, T., and Bujnicki, J.M. (2011). ModeRNA: a tool for comparative modeling of RNA 3D structure. *Nucleic Acids Res.* 39 (10): 4007–4022.
- Rouskin, S., Zubradt, M., Washietl, S. et al. (2014). Genome-wide probing of RNA structure reveals active unfolding of mRNA structures in vivo. *Nature.* 505 (7485): 701–705.
- Rupert, P.B. and Ferré-D'Amaré, A.R. (2001). Crystal structure of a hairpin ribozyme-inhibitor complex with implications for catalysis. *Nature.* 410: 780–786.
- Sabarinathan, R., Tafer, H., Seemann, S.E. et al. (2013). RNAsnp: efficient detection of local RNA secondary structure changes induced by SNPs. *Hum. Mutat.* 34 (4): 546–556.
- Salari, R., Kimchi-Sarfaty, C., Gottesman, M.M., and Przytycka, T.M. (2013). Sensitive measurement of single-nucleotide polymorphism-induced changes of RNA conformation: application to disease studies. *Nucleic Acids Res.* 41 (1): 44–53.
- Sankoff, D. (1985). Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM J. Appl. Math.* 45: 810–825.
- SantaLucia, J. Jr. (1998). A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proc. Natl. Acad. Sci. USA.* 95: 1460–1465.
- Schroeder, S.J. and Turner, D.H. (2009). Optical melting measurements of nucleic acid thermodynamics. *Methods Enzymol.* 468: 371–387.
- Schroeder, S.J., Burkard, M.E., and Turner, D.H. (1999). The energetics of small internal loops in RNA. *Biopolymers.* 52: 157–167.
- Schultes, E.A. and Bartel, D.P. (2000). One sequence, two ribozymes: implications for emergence of new ribozyme folds. *Science.* 289: 448–452.
- Seetin, M.G. and Mathews, D.H. (2011). Automated RNA tertiary structure prediction from secondary structure and low-resolution restraints. *J. Comput. Chem.* 32 (10): 2232–2244.
- Seetin, M.G. and Mathews, D.H. (2012a). RNA structure prediction: an overview of methods. *Methods Mol. Biol.* 905: 99–122.
- Seetin, M.G. and Mathews, D.H. (2012b). TurboKnot: rapid prediction of conserved RNA secondary structures including Pseudoknots. *Bioinformatics.* 28: 792–798.
- Simons, K.T., Kooperberg, C., Huang, E., and Baker, D. (1997). Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *J. Mol. Biol.* 268 (1): 209–225.
- Sloma, M.F. and Mathews, D.H. (2015). Improving RNA secondary structure prediction with structure mapping data. *Methods Enzymol.* 553: 91–114.
- Speck, M. and Lind, A. (1982). Structural analyses of *E. coli* 5S RNA fragments, their associates and complexes with proteins L18 and L25. *Nucleic Acids Res.* 10: 947–965.
- Spitale, R.C., Crisalli, P., Flynn, R.A. et al. (2013). RNA SHAPE analysis in living cells. *Nat. Chem. Biol.* 9 (1): 18–20.
- Sprinzel, M., Horn, C., Brown, M. et al. (1998). Compilation of tRNA sequences and sequences of tRNA genes. *Nucleic Acids Res.* 26: 148–153.
- Sukosd, Z., Swenson, M.S., Kjems, J., and Heitsch, C.E. (2013). Evaluating the accuracy of SHAPE-directed RNA secondary structure predictions. *Nucleic Acids Res.* 41 (5): 2807–2816.
- Szymanski, M., Barciszewska, M.Z., Barciszewski, J., and Erdmann, V.A. (2000). 5S ribosomal RNA database Y2K. *Nucleic Acids Res.* 28: 166–167.
- Tafer, H. and Hofacker, I.L. (2008). RNAplex: a fast tool for RNA-RNA interaction search. *Bioinformatics.* 24 (22): 2657–2663.
- Tafer, H., Ameres, S.L., Obernosterer, G. et al. (2008). The impact of target site accessibility on the design of effective siRNAs. *Nat. Biotechnol.* 26 (5): 578–583.
- Talkish, J., May, G., Lin, Y. et al. (2014). Mod-seq: high-throughput sequencing for chemical probing of RNA structure. *RNA.* 20 (5): 713–720.
- Tinoco, I.a. Jr. and Bustamante, C. (1999). How RNA folds. *J. Mol. Biol.* 293 (2): 271–281.
- Torarinsson, E., Havgaard, J.H., and Gorodkin, J. (2007). Multiple structural alignment and clustering of RNA sequences. *Bioinformatics.* 23 (8): 926–932.

- Turner, D.H. (2000). Conformational changes. In: *Nucleic Acids* (eds. V. Bloomfield, D. Crothers and I. Tinoco), 259–334. Sausalito, CA: University Science Books.
- Turner, D.H. and Mathews, D.H. (2010). NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic Acids Res.* 38 (Database issue): D280–D282.
- Underwood, J.G., Uzilov, A.V., Katzman, S. et al. (2010). FragSeq: transcriptome-wide RNA structure probing using high-throughput sequencing. *Nat. Methods.* 7 (12): 995–1001.
- Walter, P. and Blobel, G. (1982). Signal recognition particle contains a 7S RNA essential for protein translocation across the endoplasmic reticulum. *Nature.* 299: 691–698.
- Waugh, A., Gendron, P., Altman, R. et al. (2002). RNAML: a standard syntax for exchanging RNA information. *RNA.* 8: 707–717.
- Weinreb, C., Riesselman, A.J., Ingraham, J.B. et al. (2016). 3D RNA and functional interactions from evolutionary couplings. *Cell.* 165 (4): 963–975.
- Will, S., Reiche, K., Hofacker, I.L. et al. (2007). Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering. *PLoS Comput. Biol.* 3 (4): e65.
- Will, S., Joshi, T., Hofacker, I.L. et al. (2012). LocARNA-P: accurate boundary prediction and improved detection of structural RNAs. *RNA.* 18 (5): 900–914.
- Woodson, S.A. (2000). Recent insights on RNA folding mechanisms from catalytic RNA. *Cell. Mol. Life Sci.* 57 (5): 796–808.
- Wuchty, S., Fontana, W., Hofacker, I.L., and Schuster, P. (1999). Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers.* 49: 145–165.
- Xia, T., SantaLucia, J. Jr., Burkard, M.E. et al. (1998). Thermodynamic parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson-Crick pairs. *Biochemistry.* 37: 14719–14735.
- Xia, T., Mathews, D.H., and Turner, D.H. (1999). Thermodynamics of RNA secondary structure formation. In: *Prebiotic Chemistry, Molecular Fossils, Nucleosides, and RNA* (eds. D.G. Söll, S. Nishimura and P.B. Moore), 21–47. New York, NY: Elsevier.
- Xu, Z. and Mathews, D.H. (2011). Multilign: an algorithm to predict secondary structures conserved in multiple RNA sequences. *Bioinformatics.* 27 (5): 626–632.
- Xu, Z.Z. and Mathews, D.H. (2016). Secondary structure prediction of single sequences using RNAstructure. *Methods Mol. Biol.* 1490: 15–34.
- Zadeh, J.N., Steenberg, C.D., Bois, J.S. et al. (2010). NUPACK: analysis and design of nucleic acid systems. *J. Comput. Chem.* 32: 170–173.
- Zadeh, J.N., Wolfe, B.R., and Pierce, N.A. (2011). Nucleic acid sequence design via efficient ensemble defect optimization. *J. Comput. Chem.* 32: 439–452.
- Zuber, J., Sun, H., Zhang, X. et al. (2017). A sensitivity analysis of RNA folding nearest neighbor parameters identifies a subset of free energy parameters with the greatest impact on RNA secondary structure prediction. *Nucleic Acids Res.* 45 (10): 6168–6176.
- Zuker, M. (1989). On finding all suboptimal foldings of an RNA molecule. *Science.* 244: 48–52.
- Zuker, M. (2003). Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res.* 31 (13): 3406–3415.
- Zuker, M. and Jacobson, A.B. (1995). "Well-determined" regions in RNA secondary structure predictions. Applications to small and large subunit rRNA. *Nucleic Acids Res.* 23: 2791–2798.
- Zuker, M. and Jacobson, A.B. (1998). Using reliability information to annotate RNA secondary structures. *RNA.* 4: 669–679.
- Zuker, M. and Sankoff, D. (1984). RNA secondary structures and their prediction. *Bull. Math. Biol.* 46: 591–621.
- Zuker, M. and Stiegler, P. (1981). Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res.* 9: 133–148.

7

Predictive Methods Using Protein Sequences

Jonas Reeb, Tatyana Goldberg, Yanay Ofra, and Burkhard Rost

Introduction

Simply put, DNA encodes the instructions for life, while proteins constitute the machinery of life. DNA is transcribed into RNA and from there information is delivered into the amino acid sequence of a protein. This simplified version of the “central dogma of molecular biology” formulated by Francis Crick (1958) essentially remains valid, although new discoveries have extended our view (Elbarbary et al. 2016). Furthermore, epigenetic studies have demonstrated that chromatin contains more complex information than just a one-dimensional (1D) string of letters, with the heritability of epigenetic traits having a profound effect on gene expression (Allis and Jenuwein 2016). Nonetheless, the 1D protein sequence ultimately determines the three-dimensional (3D) structure into which the protein will fold (Anfinsen 1973), where it will reside in the cell, with which other molecules it will interact, its biochemical and physiological function, and when and how it will eventually be broken down and reduced back into its building blocks. In sum, the function (or, in the case of a disease, the malfunction) of every protein is encoded in the sequence of amino acids.

The central dogma suggests that everything about a protein can be inferred from its DNA sequence – so, why then analyze protein sequences? It turns out that computationally converting DNA to protein sequence is challenging and we still do not understand exactly how to identify the structure of a protein based on the DNA that encodes it. It is even more difficult to predict transcripts from DNA. Fortunately, many experimental approaches, including proteomics methods, can be used to deduce protein sequences, as discussed in Chapter 11.

The advent of “next-generation” DNA sequencing technologies is generating a wealth of raw sequence data about which very little is known (Martinez and Nelson 2010; Goodwin et al. 2016). The pace at which sequences are accumulating far exceeds the ability of experimental biologists to decipher their biochemical traits and biological functions. The gap between the number of proteins of known sequence and known function – the “sequence–function gap” – is ever increasing, requiring improved computational approaches to predict aspects of a protein’s function from its amino acid sequence. A similar sequence–structure gap exists for proteins, in that there are 180 million protein sequences available but only about 150 000 different known protein 3D structures have been determined as of this writing (Berman et al. 2000; UniProt Consortium 2016).

Determining a protein’s function begins with an analysis of what is already known. This means that every protein must be compared with all others, which implies that the computational time needed to study protein function grows as the square of sequence growth – a tremendous challenge for computational biology and bioinformatics. In the following sections, we survey some of the research approaches and computational tools that have been shown to successfully predict aspects of the structure and function of a protein from its amino acid sequence.

One-Dimensional Prediction of Protein Structure

Synopsis

The 1D structure of a protein can be written as a simple string of characters representing the set of natural amino acids – that is, the information content is one dimensional. While more details on protein structure can be found in Chapter 12, in this chapter, we will focus specifically on 1D prediction methods. Predictions of 1D features are relevant for two reasons. First, features such as the number of membrane helices, the disorder in a protein, or the surface residues are often important for protein function. We could determine 1D structure from 3D structures if such structures were experimentally available but, given the sequence–structure gap discussed above, experimental 3D structures are available for fewer than 1% of all known sequences, while 1D predictions can be obtained for all 180 million protein sequences known today. Second, predictions of 1D structure are used as an input for most of the methods that will be described in the section about functional prediction that follows. All of the features that will be described here are available from the PredictProtein server, illustrated in Figure 7.1 and providing pre-computed data on over 20 million proteins (Rost et al. 2004; Kajan et al. 2013; Yachdav et al. 2014).

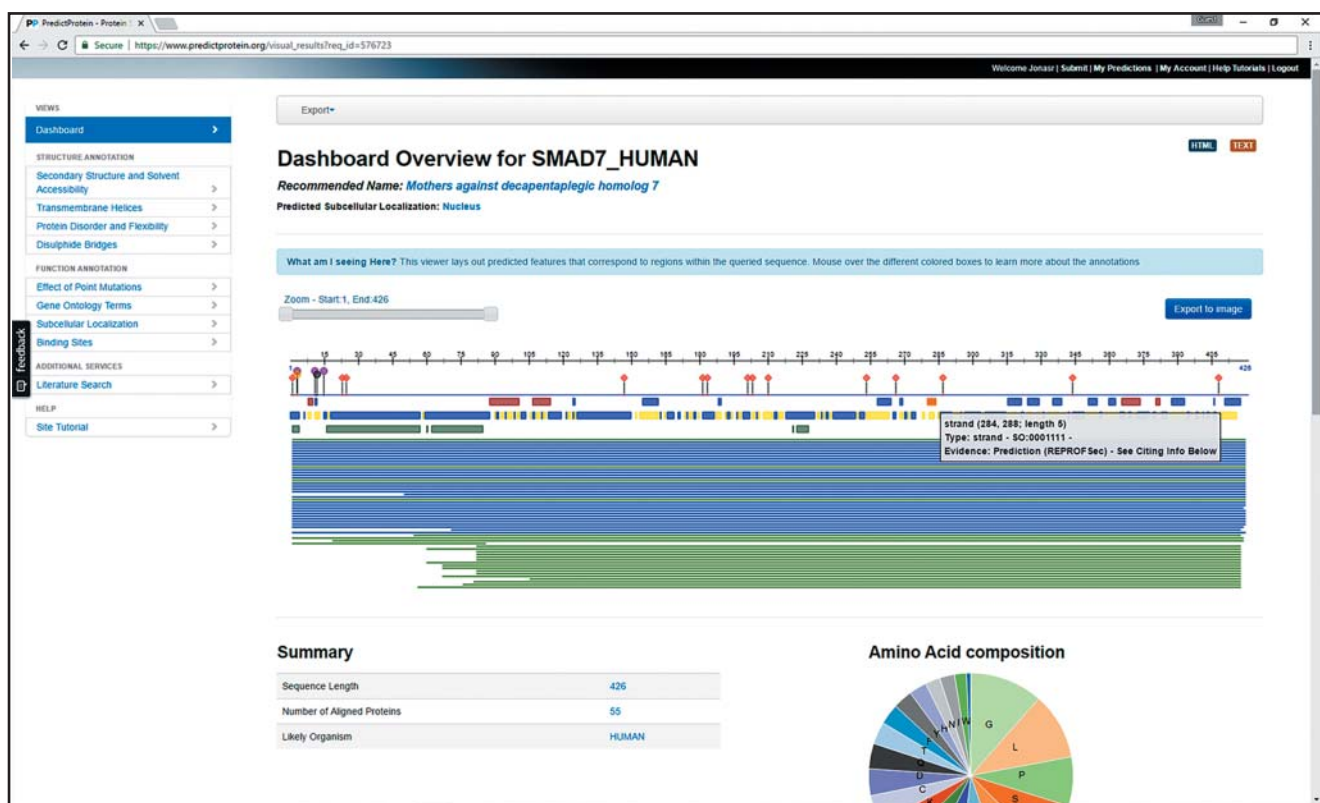


Figure 7.1 Dashboard of the PredictProtein web server. PredictProtein (Yachdav et al. 2014) provides a centralized interface to many methods that predict aspects of protein structure and function from sequence. Shown here is a sample of the dashboard for the protein picturesquely named *Mothers against decapentaplegic homolog 7* (UniProtKB identifier smad7_human). The black, numbered line in the upper middle indicates the input amino acid sequence of length 428. Below follow predictions of different sequence-based tools, along with a synopsis of the protein family. Predictions include protein–protein binding, protein–DNA/RNA binding, residue exposure (solvent accessibility), secondary structure, and residue flexibility; if found, the predictions also include membrane, long disordered, and coiled-coil regions. Additional information is shown through mouse-over events, here illustrated through the beta strand prediction from the method ReProf. Tabs on the left give access to more detailed views of various predictions and analyses.

Secondary Structure and Solvent Accessibility

Background Secondary structures are local macrostructures formed from short stretches of amino acid residues that organize themselves in specific ways to form the overall 3D structure of a protein. Physically, the driving force behind the formation of secondary structures is a complex combination of local and global forces. Secondary structure prediction methods seek to predict the secondary structure of a short protein stretch based on the sum of these forces. For instance, *alpha helices* are stabilized by hydrogen bonds between the CO group of one amino acid and the NH group of the amino acid that is four positions C-terminal to it. Strands are structures in which the backbone zigzags to create an extended structure. The most common among these is called the *beta strand*. Two or more stretches of beta strands often interact with each other, through hydrogen bonds formed between the different strands, to create a planar structure known as a *beta sheet*. Structures that are neither helices nor strands are referred to as “coils,” “others,” or “loops” (Figure 7.2).

Predicting secondary structure is an important step in the experimental study of proteins and toward inferring their function and evolution. Dozens of ideas, approaches, and methods for secondary structure prediction have been suggested over the last decades. They are based on various biochemical insights and a wide range of computational techniques. At their core, all methods identify patterns of amino acid frequencies that correlate with the formation of secondary structure elements. The first approach was based just on the occurrence of prolines, whose structure tends to interrupt helices (Szent-Györgyi and Cohen 1957). Further work was based on the clustered occurrence of residues statistically identified to be either helix- or strand-“former” residues (Chou and Fasman 1974). However, single amino acids do not inherently contain sufficient information for reliable secondary structure prediction. Therefore, the next set of methods that were developed typically employed a sliding window approach in which the secondary structure prediction for a central residue is based on the information of the surrounding residues. For example, in a sliding window of seven residues, the input consists of the central residue together with the three preceding and three following residues. The information about the residues in this window can then be extracted in several ways; for example, one could count how often a specific combination of seven residues was found in known 3D structures, then use that information to predict the secondary structure class that was most often observed in the known data. This is the approach of one of the oldest secondary structure prediction methods, called GOR (Garnier et al. 1978, 1996). A crucial extension of this concept was the incorporation of evolutionary information (Rost and Sander



Figure 7.2 Protein secondary structure. Experimentally determined three-dimensional structure of alcohol dehydrogenase (UniProtKB identifier ADHX_HUMAN) rendered in PyMOL (PDB structure 1m6h, chain A; Schrodinger 2015). The protein is shown in a “cartoon” view that highlights its secondary structure elements: alpha helices in are shown in red, beta strands are depicted as yellow arrows, and all other (“loops”) are colored in green.

1993). Typically, this is done by finding homologs of the query sequence in a large database, using position-specific iterated (PSI)-BLAST (Altschul and Gish 1996) or the more sensitive HHblits, which uses hidden Markov models (HMMs; Box 7.1) to perform sequence profile comparisons (Remmert et al. 2012). The resulting hits are then aligned in a multiple sequence alignment (MSA; Chapter 8) that, in turn, is often represented as a position-specific scoring matrix (PSSM; Chapter 3) or similar construct. As homologous proteins typically have similar structures, the substitution patterns found within the MSA contain valuable information about these proteins' secondary structure. Conserved regions likely mean that secondary structure is present at those positions, thus amino acid frequencies can be weighted by the overall conservation of residues at each position within the MSA.

Box 7.1 Hidden Markov Models

Hidden Markov models (HMMs) provide a statistical representation of real biological processes. In the context of gene prediction, HMMs describe information about the controlled syntax of the structure of a gene. In the context of protein analysis, different syntactical rules or "grammars" are used for different applications. The first and arguably the most important application of HMMs for proteins is the creation of a reliable multiple sequence alignment (MSA) for a given protein family.

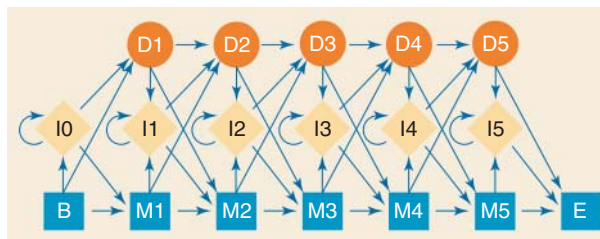
Consider a simple MSA of length six:

```
Q-WKPG
Q-WKPG
Q-WRPG
QIWK-G
Q-WRPG
Q-WRPG
```

Some of the positions, such as the first glutamine (Q), are absolutely conserved, while position 4 is occupied only by positively charged residues. Position 2 shows a gap introduced by an insertion in one sequence; position 5 shows a gap introduced by a deletion in the same sequence.

Each of these observations can be represented by different states in the HMM. The match state represents the most probable amino acid(s) found at each position of the alignment; match is a bit of a misnomer, because the match state considers the probability of finding a given amino acid at that position of the alignment. If the position is not absolutely conserved, the probabilities are adjusted accordingly, given the residues found in that column. The insert state represents the case when an additional amino acid needs to be accommodated, as it is here with the isoleucine (I) in sequence 4. The delete state represents the case when an amino acid is missing and when the sequence (here, sequence 4) must jump over a position to continue the alignment.

Each of these states, as well as the relationship between the states, can be illustrated graphically using a commonly used representation originally introduced by Anders Krogh:



Here, the lower row represents the match states, with B representing the beginning of the alignment and E the end of the alignment. Each of the diamonds represents the insert states, and each of the circles represents the delete states. Arrows represent a movement from state to state, and each is associated with the probability of moving from one state to another.

Returning to the alignment, M1 would require a glutamine, M2 a tryptophan, M3 either a lysine or an arginine (50% of the time for either), M4 a proline, and M5 a glycine. Given this, to represent the sequence EWRPG, the movement through the model would take the form $B \rightarrow M1 \rightarrow M2 \rightarrow M3 \rightarrow M4 \rightarrow M5 \rightarrow E$, because the sequence does not require an insertion or a deletion to align with the rest of the sequences in the group, with the exception of sequence 4. The path for sequence 4 obviously would be different than for most of the sequences; it would take the form $B \rightarrow M1 \rightarrow I1 \rightarrow M2 \rightarrow M3 \rightarrow D4 \rightarrow M5 \rightarrow E$. The movement from M1 to I1 accounts for the insertion at position 2, and the movement from M3 to D4 accounts for the deletion at position 6.

The usefulness and elegance of this model comes from the ability to train the model based on a set of sequences. Returning to the original sequences, imagine that they were fed into the model one by one, without knowing the alignment in advance. For each sequence, the most probable path through the model is determined. As soon as this is carried out for all the sequences, the transition probabilities and probabilities for each match state are determined. This, in turn, can be used to generate the best alignment of the sequences. Knowledge of these probabilities also allows for new sequences to be aligned to the original set.

Using the collective characteristics of the input sequences either allows these profile HMMs to be used to scan databases for new sequences belonging to the set or individual sequences can be scanned against a series of HMMs to see whether a new sequence of interest belongs to a previously characterized family.

Another important feature of a single residue is its solvent accessibility or accessible surface area (ASA); this is the area of a residue's surface that is exposed to the solvent and that could interact with other proteins or smaller molecules (Figure 7.3). After the elucidation of the first protein structure, Lee and Richards started examining the ASAs of proteins by developing a program to draw van der Waals radii on the protein's structure, essentially by "rolling a ball" over the entire surface of the protein (Lee and Richards 1971). The accessibility is then defined by the protein's surface area, typically given in angstroms squared, that can be touched by a water molecule's radius. Biologically, the concept of accessibility is of interest, as residues deeply buried inside a protein may be crucial in stabilizing its structure even though they cannot partake directly in binding other molecules. Many methods that perform secondary structure prediction also predict accessibility to aid the identification of active residues, which is important for functional studies.

The secondary structure annotations and solvent accessibility data needed for training the prediction methods can be obtained from known 3D protein structures. The most popular tool for this task, the Dictionary of Secondary Structure for Proteins (DSSP; Kabsch and Sander 1983), analyzes hydrogen bonds in 3D structures and assigns eight types of secondary structure elements that can be grouped into the previously mentioned classes: helices (alpha, 3_{10} , and pi helices), strands (extended and bridge), and other (usually referred to as "turn," "bend," and "other" within the DSSP). Solvent accessibility is given for each residue in angstroms squared. Other similar programs include STRIDE, which incorporates backbone geometry information and aims to provide annotations that are consistent with those provided by experimentalists (Frishman and Argos 1995; Heinig and Frishman 2004), and NACCESS, which calculates solvent accessibility from structures (Hubbard and Thornton 1993).

We now describe the most popular secondary structure and solvent accessibility prediction servers. Nearly all of today's methods predict just three types of secondary structure (helix, beta strand, and other) and numerical (0–9) or categorical (e.g. buried, partially buried, or exposed) measures of accessibility.

Methods PHDsec is among the earliest methods applying the idea of machine learning to the prediction of secondary structure (Rost and Sander 1993). It was the first method that combined the usage of evolutionary information with machine learning. This information is

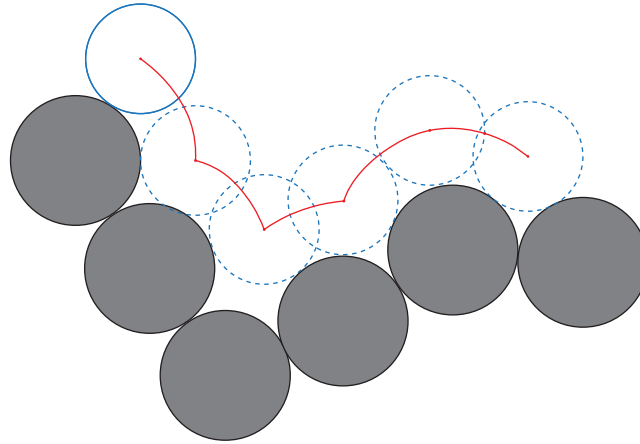


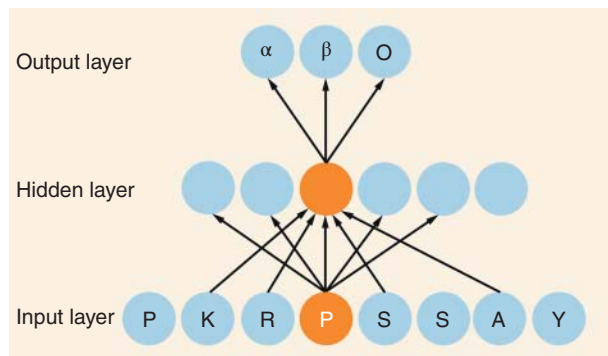
Figure 7.3 Accessible surface area (ASA). The ASA describes the surface that is in direct contact with the solvent. Practically, the solvent is usually water, and the ASA can be defined by rolling a probe over the surface of the protein. The probe is a sphere representing the size of a water molecule (blue), while the protein's surface is defined by the van der Waals volume of every amino acid's atoms (gray). The ASA is then described by the center of the probe as it moves along the surface (red). Software such as DSSP that calculate ASA employ more efficient algorithms; however, the underlying principle of what is being measured remains the same.

obtained through the Homology-derived Secondary Structure of Proteins (HSSP) database (Touw et al. 2015), which contains alignments of protein sequences to those of proteins with known structures contained in the Protein Data Bank (PDB) (Rose et al. 2017). PHDsec uses two consecutive neural networks to make its predictions (Box 7.2): the first layer uses conservation values derived from an MSA within the HSSP database, using a window of 13 residues to predict one of three secondary structure states. The first layer's output is then used as input to the second layer, which smooths out biologically implausible predictions, such as an alpha helix that is interrupted by a single beta sheet residue in the middle of the helix. Method improvements since the first version include the ability to factor in additional weighting information derived from MSAs or global amino acid composition features (Rost and Sander 1994a,b). This process also led to the renaming of PHDsec as PROFsec. The most recent iteration, ReProf, is accessible as part of the PredictProtein web server (Yachdav et al. 2014). ReProf also includes the most recent version of the solvent accessibility prediction algorithm, initially named PHDacc and PROFacc (Rost and Sander 1994a,b).

Box 7.2 Neural Networks

With the development of deep learning, neural networks have regained their popularity as machine learning models for biology (Punta and Rost 2008; Jensen and Bateman 2011). Neural networks attempt to mimic the way the human brain processes information when trying to make a meaningful conclusion based on previously seen patterns. The type of network design that is typically used for machine learning has several layers: an input layer, zero or more hidden layers, and one output layer. The figure shows an example with one hidden layer.

Here, the input layer is a protein sequence (PKRPSSAY), and the output layer is one of the possible outcomes: whether a particular amino acid lies within an alpha helix, a beta strand, or other (loop) region. The neural network receives its signals from the input layer and passes information to the hidden layer through a neuron, similar to how a neuron would fire across a synapse. A representative subset of the neurons is shown here by arrows. In most applications all input units are connected to all hidden, and all hidden to all output units.



In the figure, a proline residue is highlighted in the input layer; it influences several nodes in the hidden layer, the strength or weight of the input being controlled by the neurons. In the hidden layer, one of the nodes is shown in orange; many of the positions in the sequence may influence that node, with the neurons again controlling the degree to which those inputs influence the node. The figure illustrates a feed-forward neural network, where the flow of information is in one direction. Recurrent neural networks, where neurons connect back to neurons in earlier layers, are also possible.

Why use a neural network? If the direct relationship between the input and output layer was perfectly understood, such a complicated approach would not be necessary, because one could write cause-and-effect rules that would make the intermediate, hidden layer(s) unnecessary. In the absence of this direct relationship, neural networks can be used to deduce the probable relationship between the input and output layers; the only requirement is the knowledge that the input and output layers are, indeed, related. Here, there is an obvious relationship between the input and output layers, because the individual amino acids found within a protein must belong to one of the three given secondary structure classes modeled by the output layer.

To start deducing the relation between input and output, a supervised learning approach is used, i.e. the connections are optimized to fit a set of training examples for which the input→output mapping is known. For example, in the realm of secondary structure prediction, one would construct datasets based on known 3D structures, noting not only the secondary structure in which a particular residue is found but also other factors influencing structural conformation. Based on these training data, the network attempts to learn the relationship between the input and output layers, adjusting the strength of each of the interconnected neurons to fine-tune the predictive power of the network.

PSIPRED is another neural network-based secondary structure predictor using an approach similar to PHDsec (Jones 1999). A profile based on the initial query sequence is created using PSI-BLAST and then fed to the first neural network using a window size of 15. The output is then further processed using a second network and the same window size. The network architecture currently used for performing these predictions has been significantly improved over time (Buchan et al. 2013).

Proteus directly transfers secondary structure annotation information obtained from homologs of the protein being analyzed into the prediction pipeline (Montgomerie et al. 2006). A BLAST search is performed in order to find homologs of known structure within PDB that match the query protein. If successful, the secondary structure annotation of the homolog found in PDB is copied onto all aligned residues of the query protein. Any non-aligned residues are then annotated using predictions based on the query sequence itself. Once this process is complete, three methods (including PSIPRED) are executed and their output is fed into a neural network that yields a consensus prediction that is finally reported.

SANN is a solvent accessibility predictor (Joo et al. 2012). Using a set of proteins with known accessibility, the authors used PSI-BLAST-based PSSMs with a sliding window of size 15 as feature vectors. The vector of a residue in the query protein is then compared with all vectors in the database, with the central residue having the highest weight and decreasing outwards. Based on this, the 64 nearest neighbors are then used to perform a prediction based on a z -score that weights neighbors by their closeness to the query residues. SANN predicts solvent accessibility in two (buried, exposed) or three (buried, intermediate, exposed) discrete states. It also gives a fractional prediction for relative solvent accessibility (RSA) between 0 and 1.

SSpro5 predicts secondary structure in three or eight (“SSpro8”) states. It can make predictions directly, based on the query sequence, or based on homology of the query sequence to proteins of known structure, similar to the approach used by the Proteus method (Magnan and Baldi 2014). First, BLAST is used to identify query protein sequence matches to known structures in PDB. Once found, the most common secondary structure class assigned by DSSP at each known structure residue is output as the prediction. If this fails (e.g. when BLAST finds no matches to PDB or there is no most-common secondary structure class), a sequence-based prediction is used. For this, a set of 100 bidirectional recurrent neural networks (BRNNs) has been trained, using PSI-BLAST PSSMs as its input (Box 7.2). The bidirectional neural network architecture enables the predictor to consider information from a central sequence window, as well as sequence regions occurring before and after the central window (Pollastri et al. 2002). ACCpro5 uses the same methodology to perform predictions of solvent accessibility in two states (RSA larger or smaller than 25%) or 20 states (RSA 0–95% in 5% increments).

RaptorX Property is part of a larger structure prediction service and predicts secondary structure, solvent accessibility, and disorder (Wang et al. 2016a,b) (Figure 7.4). The method

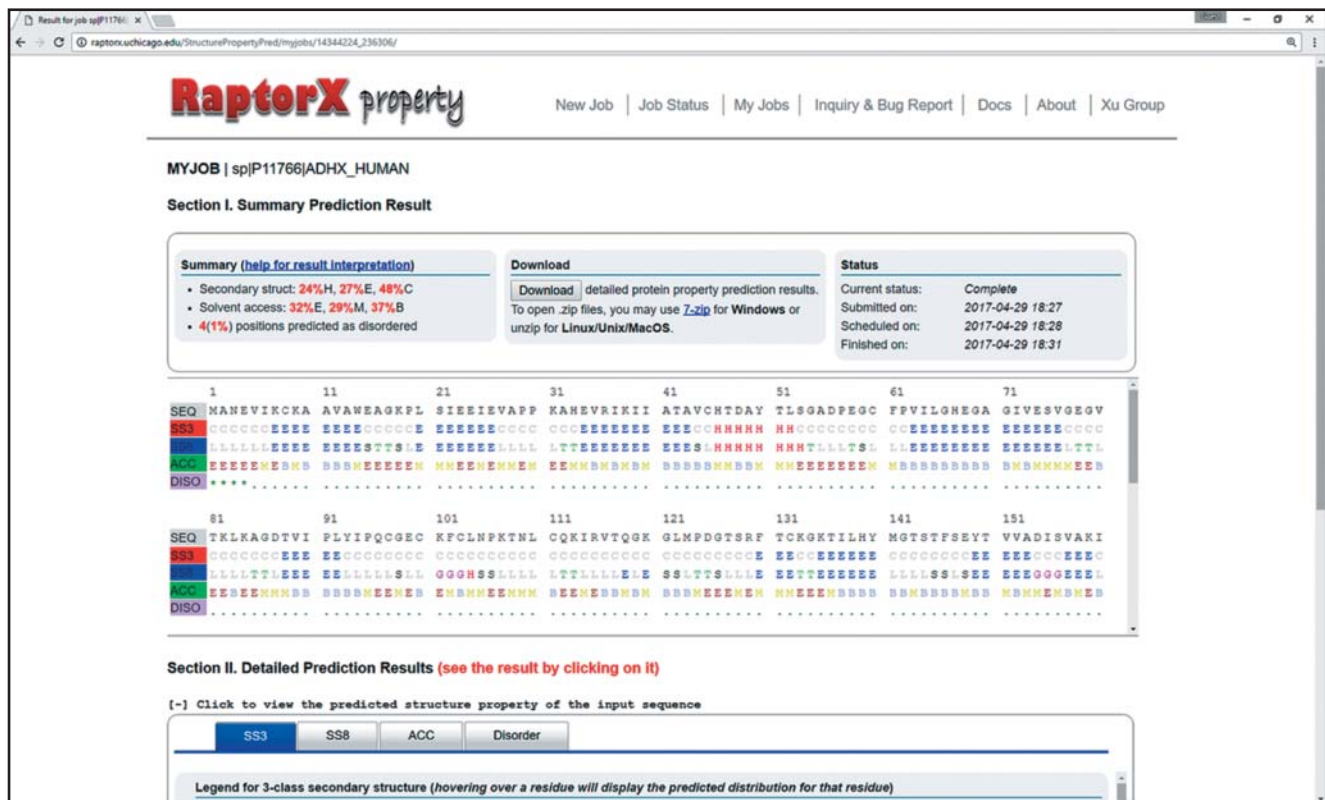


Figure 7.4 Protein secondary structure. Prediction of secondary structure, solvent accessibility, and disordered regions of the same protein by the web server RaptorX Property. Summary statistics about the prediction results are shown at the top left. Below, the first row contains the input amino acid sequence, followed by the predicted secondary structure in the three DSSP classes (SS3) followed by the eight DSSP classes (SS8). The last two rows contain solvent accessibility prediction in three states (exposed, medium, and buried) and a binary prediction of disordered residues (none detected here).

also uses a deep learning approach called deep convolutional neural fields, using sequence profiles as input. This deep learning approach is expected to capture both relevant aspects of global protein structure and observed correlations with neighboring residues. The network achieved its best performance using a window size of 11 and seven hidden layers. Disordered residues (discussed in more detail in Disordered Regions) and solvent accessibility (either buried, medium, or exposed) are predicted using the same approach.

SPIDER3 is a deep learning-based method that predicts secondary structure, solvent accessibility, and backbone torsion angles, as well as two other angles involving the $C\alpha$ atoms and neighboring residues (Heffernan et al. 2017). The correlation between these three elements is exploited by an iterative prediction approach. In the first iteration, the various structure properties are predicted separately using two BRNNs. In the following iterations, the output from one of these BRNNs is used as input for the other to iteratively predict each output structure property in turn based on all other structural properties. The input in the first iteration, to which intermediate predictions are then added in later iterations, consists of just seven physicochemical amino acid properties, together with sequence profiles from PSI-BLAST and HHblits. Owing to the design of the network, the method is not window based but uses the whole sequence as input.

Performance Assessment of Secondary Structure Prediction

The most common measure for the assessment of secondary structure prediction is called the Q_3 score (Box 7.3). Additional measures should also be considered because of the imbalance of the three possible states (alpha helix, beta strand, or random coil). Most methods predict strands least accurately owing to inherent bias in the experimental training data (Rost 1996, 2001); only about 20% of all residues are in beta strands, over 30% are in helices, and about 50% in other structures. Many methods that reach relatively high Q_3 scores perform as badly as random when predicting beta strands. Another aspect, considered by per-segment scores, is how well an entire secondary structure segment is predicted. The most prominent example of such a score is the segment overlap score (SOV), which measures correctly predicted residues and segments (see Box 7.4) (Fidelis et al. 1999).

Box 7.3 Secondary Structure Prediction Scoring Schemes and Receiver Operating Characteristic Curves

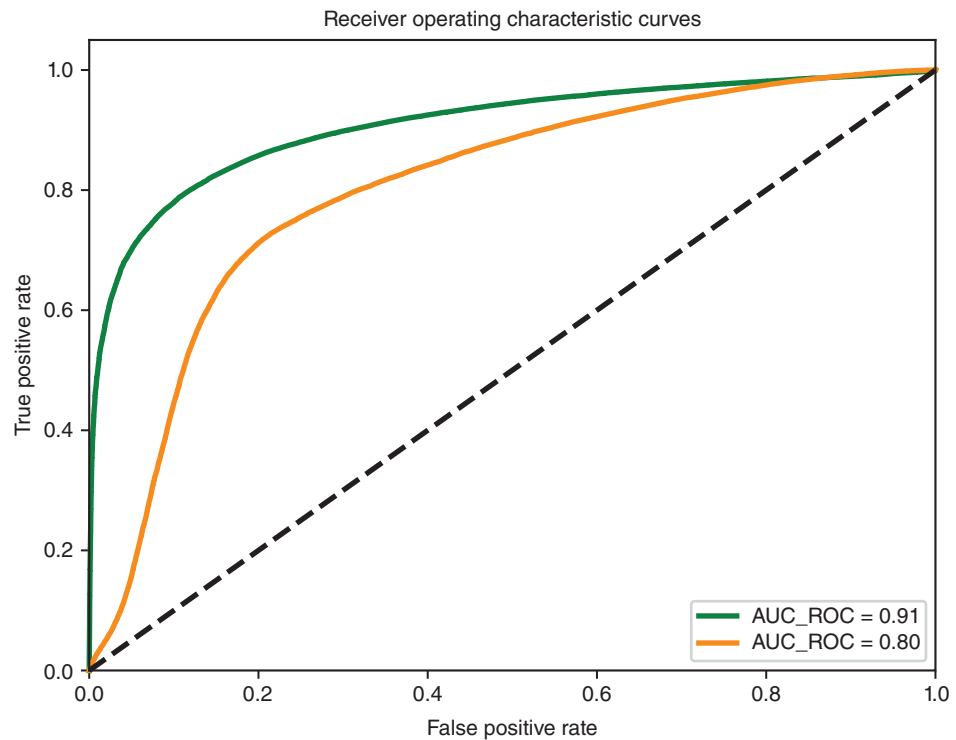
A classification task can discern between two or more classes. For example, a secondary structure prediction typically yields a per-residue classification of either an alpha helix, beta strand, or non-structured region. Measures of classification performance, such as the number of true positives, can differ in meaning between publications; thus, one must carefully identify the relevant definition within the publication. Most commonly, multi-class prediction performance is reported using accuracy (see Box 5.4), which intuitively translates to multiple classes when its definition is regarded as “the number of correct predictions among *all* predictions.” For some prediction tasks such as secondary structure prediction, this score has also been termed Q_n , with n denoting the number of classes. As secondary structure predictions ultimately result in the detection of one of three classes, this metric is often called the Q_3 score.

Binary classification usually treats prediction results as discrete. However, machine learning models usually output a continuous number, representing a likelihood for that class. During development of the method, a threshold is chosen that yields the best discrimination between the classes. Imagine a simple neural network with one output node that provides a continuous value between 0 and 1. Further, this network was trained with examples of residues residing inside the membrane (producing a desired output of 1) and with residues residing outside the membrane (producing a desired

(Continued)

Box 7.3 (Continued)

output of 0). Given a query residue, the network will now predict a value between 0 and 1; the closer that value is to 1, the more likely the residue is found inside the membrane. The receiver operating characteristic (ROC) curve is used in these cases to describe how the choice of threshold for the predictor output affects the true-positive rate (TPR) and the false-positive rate (FPR; see Box 5.4).



This curve is derived by making predictions for every possible decision threshold, then plotting the resulting TPR and FPR values. The plot forms a curve that describes the discrimination performance of the prediction method. A perfect predictor would have a TPR of 1 and FPR of 0 across all relevant thresholds, while a random prediction is represented by the dashed black line. Instead of displaying the ROC curve itself, one can represent the curve as a single numerical score, called “the area under the curve” (AUC). Typically, AUC denotes the area under the ROC curve (AUC_ROC); however, this term is also used to describe other areas, such as the area under the precision vs. recall curve (AUC_PR).

Recent methods estimate Q_3 and SOV values around 80–85% for their prediction performance (Mirabello and Pollastri 2013; Heffernan et al. 2015; Wang et al. 2016a); those values have been confirmed in a recent review for a small set of methods (Yang et al. 2016a). These measures also seem realistic given the increasingly slow advances in the development of new methodologies since the last independent evaluation published several years before this writing, where only small performance increases were reported (Zhang et al. 2011). Because of known errors in structure determination and resulting inconsistencies in secondary structure assignments from experimentally determined protein 3D structures, among other issues, a perfect Q_3 of 100% is not attainable (Kihara 2005; Heffernan et al. 2017). Arguably, the most important challenge in this field has shifted from predicting secondary structure to the higher goal of predicting tertiary structure (Chapter 12).

For solvent accessibility, scoring schemes differ as widely as the values that are predicted. To calculate RSA, a good normalization scheme is necessary, but this is not trivial (Tien et al. 2013). Given that categorical values such as a prediction in two states (buried or exposed) are used, Q_2 or Q_3 measures can be used in a fashion analogous to how these measures are applied to secondary structure prediction. For predictions of RSA that fall into a range such as 0–100, accuracy measures are not meaningful, and scores such as Pearson's correlation coefficient are more appropriate. From the methods described above, SANN and RaptorX Property both report Q_3 values of 66%, while SANN reports a Q_2 value of 81% and ACCpro reports a Q_2 value of 88%. Pearson's correlation coefficients for RSA prediction are 0.68 for SANN and 0.8 for SPIDER3. Unfortunately, these values cannot be directly compared, as they were computed using different training data or using different predicted output category definitions. Such decisions can impact performance since residues with 0% solvent exposure are much easier to predict than those with, for example, 36–49% exposure.

In the past, the web servers EVA and Livebench provided automated predictor performance assessments by comparing prediction results with about-to-be-released experimentally determined protein structures (Eyrich et al. 2001; Rychlewski and Fischer 2005), but, unfortunately, this service is no longer active and no independent large-scale assessment of secondary structure prediction currently exists. The Critical Assessment of Protein Structure Prediction (or CASP) is a biannual community-led challenge in which developers submit predictions for structures that have been solved but not yet publicly disclosed (Moult et al. 1995). At the time of submission, none of the structures are known to either the organizers or any of the participating groups, making CASP a double-blind experiment. Once the structures are released, assessors evaluate the previously submitted predictions to chart progress in the field in an unbiased manner. The focus of the challenge is on developing new, more accurate algorithms for protein tertiary structure prediction. In CASP5, the last iteration that considered secondary structure predictions (in 2002), method performance had approached a saturation point (Aloy et al. 2003). Given this likely saturation of method performance and because of the generally very high performance available, the choice of predictor is, to some degree, a matter of personal preference. Any recent method (such as RaptorX Property or SPIDER3) will provide high-performance predictions through easy-to-use web servers. Other services, such as ReProf and PSIPRED, are part of larger web-based suites that allow the user to run additional prediction tools on the query protein with the click of a button and are thus well suited to give an overview of a wide range of protein features.

Transmembrane Alpha Helices and Beta Strands

Background The communication between a cell and its surroundings takes place almost exclusively through proteins that are embedded in the cell membrane, with these *transmembrane* proteins interacting with molecules on both the intracellular and the extracellular sides of the membrane. Transmembrane proteins have been estimated to make up 20–30% of all proteins found in any organism (Stevens and Arkin 2000; Liu and Rost 2001; Fagerberg et al. 2010). These include well-known and highly studied protein classes such as the G-protein-coupled receptors, proteins that are often major targets of drug development efforts (Jacoby et al. 2006; Overington et al. 2006). In fact, almost two-thirds of all drug targets are transmembrane proteins. The ability to identify transmembrane proteins and to decipher their molecular mechanisms is, therefore, of high interest in many fields of biomedicine. Unfortunately, experimental structural determination of membrane-bound proteins is significantly more difficult than for soluble proteins, and transmembrane protein structures are strongly under-represented in PDB (Kloppmann et al. 2012). Therefore, computational predictions are essential for understanding the structures of this class of proteins. Typically, the transmembrane segments are classified into one of two classes according to their secondary

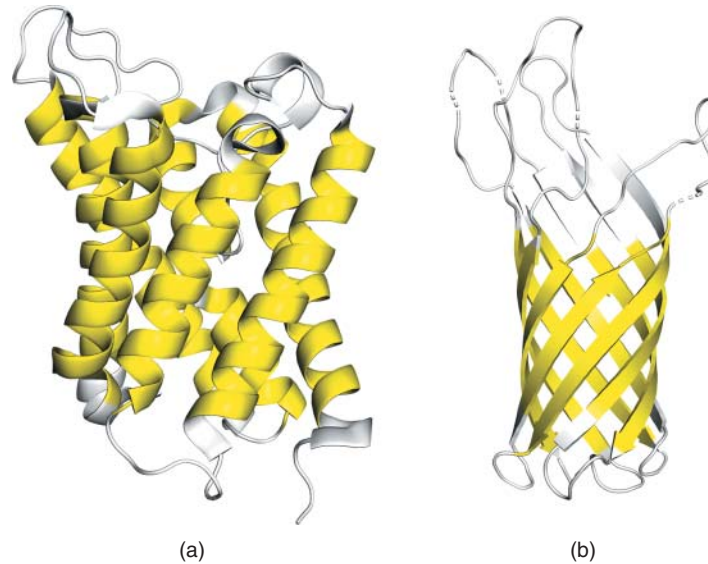


Figure 7.5 Types of transmembrane proteins. Experimentally determined three-dimensional structures of two transmembrane proteins rendered in PyMOL (Schrodinger 2015). Protein segments that are annotated as being inside the membrane are highlighted in yellow. (a) Alpha helical transmembrane protein aquaporin (PDB structure 3llq, chain A). (b) Beta barrel transmembrane protein OmpA (PDB structure 1bxw, chain A).

structures: helices or strands. Usually, the proteins consist only of these two secondary structure elements; however, both fulfill the same task of masking the polar protein backbone from the hydrophobic membrane environment. Proteins using alpha helical transmembrane segments for this purpose are much more common, while those consisting of beta strands are typically porins found only in the outer membrane of Gram-negative bacteria, mycobacteria, chloroplasts, and mitochondria (Kessel and Ben-Tal 2011) (Figure 7.5). As membrane proteins consist of the same types of secondary structure discussed in the previous section, one may wonder why specialized predictors were developed for use with transmembrane proteins rather than just using already available predictors. The underlying reason is that transmembrane proteins have evolved unique structural properties that allow them to be firmly embedded in the cell membrane. These physicochemical properties are different enough from those of soluble proteins that specialized predictors are required. Fortunately, these properties are easier to identify, making it easier to predict transmembrane segments compared with predicting “general” secondary structure. The basic biophysical property that is responsible for a residue to be buried within a membrane is *hydrophobicity*, the property that enables most of the transmembrane segments to remain within the membrane and avoid exposure to the solvent on either of its sides. Hence, the first transmembrane region prediction methods focused on a search of long hydrophobic stretches of sequence (Kyte and Doolittle 1982) and hydrophobicity metrics remain a crucial input feature for today’s most advanced methods (Reeb et al. 2014). Additionally, many methods predict the transmembrane segment *topology*, providing information regarding the orientation of helices or beta strands with respect to the cytoplasmic and non-cytoplasmic sides of the membrane. An important concept underlying the determination of topology is the “positive-inside rule,” which describes the observation that the loops on the cytoplasmic side of the membrane typically contain more positively charged residues than those on the non-cytoplasmic side (Von Heijne and Gavel 1988). Early transmembrane region prediction methods focused on the combination of window-based searches for hydrophobic stretches and the analysis of electrical charges, leading to the first widely used prediction method (Von Heijne 1992; Claros and Von Heijne 1994). Since then, and with the advent of machine learning approaches, methods to determine the structure of membrane proteins have made immense advances, as described below.

Methods Phobius, like its predecessor TMHMM, predicts transmembrane segments using an HMM that is based on representations of globular domains, loops, helix ends, and the transmembrane helix core (Krogh et al. 2001). With respect to TMHMM, Phobius adds another set of states that model signal peptides (Käll et al. 2004) because signal peptides contain a stretch of hydrophobic residues that can easily be mistaken for transmembrane segments. By combining the use of known transmembrane segments and signal peptides in one model, Phobius is capable of accurately distinguishing between these two classes. **PolyPhobius** is an extension of Phobius that keeps the HMM unchanged but achieves a better performance by using a decoding algorithm that harnesses evolutionary information from MSAs of the query sequence (Käll et al. 2005).

Proteus-2 is an extension of the original Proteus method described above for secondary structure prediction (Montgomerie et al. 2008). The method first predicts signal peptides, then checks these predictions against a database of experimentally annotated signal peptides. Next, if a query protein is homologous to one with known transmembrane segments, these are transferred to the query. If this fails, transmembrane segments are predicted using a combination of TMHMM and TMB-HUNT (Garrow et al. 2005). Finally, any unassigned residues are assigned a secondary structure class by homology-based inference or, if not possible, as predicted by Proteus.

MEMSAT-SVM consists of four separate support vector machines (SVMs) for the prediction of transmembrane helices, signal peptides, loops, and re-entrant helices (Nugent and Jones 2009). This last class is a special case of helical transmembrane segments in which a helix enters and exits the membrane at the same side (Viklund et al. 2006; von Heijne 2006). MEMSAT-SVM is one of the few methods that accurately models this case.

TMSEG is a recent method that combines machine learning with empirical filters (Bernhofer et al. 2016). First, a random forest model predicts scores for each input residue to be either in the membrane, a signal peptide, or a loop. Scores are then smoothed and continuous protein sequence positions that are consistently high scoring for a given class (e.g. transmembrane) are identified as a protein sequence segment of that class (e.g. a transmembrane region). Next, a neural network refines the previously predicted segments and, finally, another random forest model predicts the topology of the protein (Figure 7.6). A strength of this method is that it can accurately distinguish between proteins with and without membrane helices.

BETAWARE focuses on predicting whether a query protein is a transmembrane beta barrel (TMBB) protein (Savojardo et al. 2013), an understudied class of transmembrane proteins. This is achieved using a neural network with an “N-to-1” encoding that supports the input of a

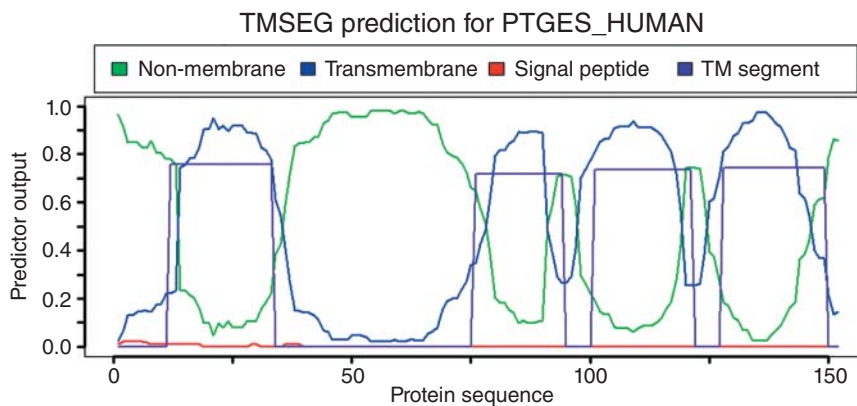


Figure 7.6 Transmembrane helix prediction by TMSEG. TMSEG (Bernhofer et al. 2016) predictions of transmembrane helices for prostaglandin E synthase (UniProtKB identifier PTGES_HUMAN). The graph distinguishes non-membrane residues (green), helical transmembrane residues (raw prediction in blue), and signal peptides (red; here all close to 0). The purple line marks the final prediction of transmembrane helices after smoothing and refinement by a segment-based neural network classifier. In this case, all four helices are correctly identified.

variable length query sequence into the network, which predicts whether the query sequence is a TMBB protein. For all putative TMBB proteins, a grammatical restrained hidden conditional random field, similar to an HMM, is used to predict the state of each residue in the protein and the protein membrane topology (Fariselli et al. 2009).

BOCTOPUS2 (Hayat et al. 2016) also predicts TMBB strands and topology. First, an SVM predicts the preference for each residue to be either pore facing, lipid facing, outer loop, or inner loop. Next, the region of the beta barrel is determined by finding the region most consistent with the predicted pore- and lipid-facing residues. Once set, the predicted classes are adjusted such that no membrane residues are predicted outside the barrel region. Lastly, an HMM that models the same four states as the SVM is used to predict the final topology.

Performance Prediction performance of transmembrane segments can be measured on three different levels: per residue, per segment, or per protein (Chen et al. 2002) (Boxes 7.3 and 7.4; see also Box 5.4). Most methods reach Q_2 scores of 80% or more (Reeb et al. 2014). However, owing to disadvantages of per-residue scores (Box 7.4) per-segment scores are important to consider when interpreting membrane prediction results. If we consider a transmembrane helix to be correctly predicted if the endpoints deviate by no more than five residues from known true-positive examples, then TMSEG and MEMSAT-SVM reach segment recall and precision values of around 85% and PolyPhobius around 75% (Bernhofer et al. 2016). On a more stringent per-protein score, Q_{ok} , TMSEG reaches 66%, MEMSAT-SVM 61%, and PolyPhobius 54%. These scores have high variance because of the small size of the independent benchmark used to compute them ($n = 44$), so these scores should be used as a general guideline instead of a perfect measure of the expected performance of each method. Similarly, performance assessment for TMBB protein prediction methods is challenging owing to a small independent benchmark size. The BOCTOPUS2 publication reports a Q_3 (membrane, inside, or outside loop) score of 89% and SOV score (Box 7.4) of 94% for their own method, and 74% Q_3 , 75% SOV for BETAWARE. These numbers are roughly confirmed in another recent publication (Tsirigos et al. 2016). Given this suggestion of higher performance along with the fact that BETAWARE is only available as a command line tool, we recommend BOCTOPUS2 as a good starting point for the prediction of transmembrane beta strands and TMBB proteins.

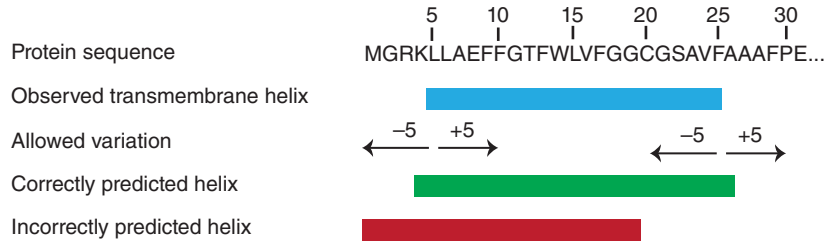
Box 7.4 Scoring Schemes for Structural Protein Segments

A disadvantage of per-residue scores such as Q_2 or Q_3 is that they do not punish biologically meaningless predictions. For example, predicting every transmembrane helix with a single non-membrane residue in its middle would not have a large effect on Q_2 , but this outcome does not make biological sense, so should therefore be penalized numerically. Per-segment scores account for this possibility, and both precision and recall (see Box 5.4) can also be measured on a per-segment basis for every protein.

Names	Formula
Segment recall, Q_{tmh}^{obs}	$\frac{\text{Number of correctly predicted transmembrane helices}}{\text{Number of observed transmembrane helices}}$
Segment precision, Q_{tmh}^{pred}	$\frac{\text{Number of correctly predicted transmembrane helices}}{\text{Number of predicted transmembrane helices}}$

To calculate these scores, one needs to define what constitutes a correctly predicted structural segment. For example, the observed and predicted positions for a helix should roughly encompass the same region of a protein. It is crucial to predict a continuous segment, while exact matches to the start and end position are not as important, as these start and end points can differ even between almost identical homologs. The problem

becomes much more complex when dealing with transmembrane proteins, as it is not at all trivial to exactly determine the beginning and end of a membrane segment even when a high-resolution 3D protein structure is available. The figure below shows the C-terminal residues of aquaporin (see Figure 7.5a), including its first transmembrane helix, experimentally determined to range from residue positions 5 to 25. Black arrows indicate the range within which a predicted helix would be considered correct if deviations of up to five residues are allowed. The same concept is employed by the segment overlap score (SOV), which measures the average overlap between the predicted and observed structural segment (Fidelis et al. 1999).



Given the per-segment scores, one can define an additional (per protein) score that measures the fraction of proteins where every transmembrane helix was correctly predicted:

$$Q_{ok} = \frac{1}{N} \sum_{i=0}^N \delta_i; \delta_i = \begin{cases} 1, & \text{if } Q_{tmh}^{obs}(i) = Q_{tmh}^{pred}(i) = 1 \\ 0, & \text{else} \end{cases}$$

Here, N is the number of proteins in the respective dataset. Clearly, this provides a much stricter cut-off than just predicting the majority of residues correctly. However, it is crucial to apply such stringency, as missing a single membrane segment can reverse the overall membrane segment topology of the protein, the orientation of which provides important clues about protein function. For example, all G-protein-coupled receptors have seven membrane helices, with the N-terminus typically residing outside of the cell (Coleman et al. 2017).

Disordered Regions

Background Disordered regions in proteins do not adopt a well-defined structure in isolation but fluctuate in a large conformational space (Habchi et al. 2014; Wright and Dyson 2014). Disordered regions are experimentally determined, for example using nuclear magnetic resonance spectroscopy, which allows the dynamic observation of a protein in solution (Habchi et al. 2014). The phenomenon of *protein disorder* or *disordered regions* is described by many different terms in the literature; these include *intrinsically unstructured*, *natively unstructured*, *natively disordered*, and *loopy*. Proteins with such regions are often referred to as intrinsically disordered proteins (IDPs) or intrinsically unstructured proteins. Typically, these intrinsically disordered regions (IDRs) have low sequence complexity and an amino acid composition bias toward hydrophilic residues.

Disordered regions have many functions. One feature is to cover a larger set of potential binding modes, i.e. a more flexible region can be induced to fit to many different shapes. Consequently, many IDPs are involved in regulation or cell signaling and appear as hubs with many binding partners in the interaction network. This is especially true for eukaryotic IDPs, while those in prokaryotes are often involved in longer lasting interactions that form complexes (van

der Lee et al. 2014). Predicting IDPs and disordered regions from sequence is of value since it can help in structure determination, drug design, and disease risk evaluation (Deng et al. 2015).

Three different approaches have been applied to predict IDRs (van der Lee et al. 2014). The first uses sequence patterns such as the amino acid composition and physicochemical properties. The second uses sequence and MSA information with machine learning. The third uses a consensus predictor strategy to combine the predictions of other tools. Protein 3D structure may also be used with any of these approaches.

All large-scale estimates for the abundance of disorder in genomes are based on prediction methods. On average, eukaryotes have more IDRs longer than 30 residues (33% of proteins) than prokaryotes (4% of proteins) (Schlessinger et al. 2011; van der Lee et al. 2014). However, individual organisms differ a lot, as does the disorder content of proteins within an organism (Habchi et al. 2014; van der Lee et al. 2014; Lobanov and Galzitskaya 2015). Furthermore, disorder correlates with the extremity of an organism's habitat (Habchi et al. 2014; Vicedo et al. 2015).

Several databases store disorder protein region information. DisProt contains manually curated, experimentally validated IDP annotations (Piovesan et al. 2016). IDEAL stores IDRs that become ordered upon protein binding and includes interaction partners of the disordered proteins (Fukuchi et al. 2014), while D²P² provides precompiled disorder predictions of nine different methods on over 1700 genomes (Oates et al. 2013).

Methods PrDOS predicts IDRs using machine learning in two parts. First, an SVM is supplied with information from a PSI-BLAST-generated PSSM in a 27-residue sliding window (Ishida and Kinoshita 2007). Second, any query homologs with known 3D structures are identified. The more aligned residues in structures are disordered, the higher the assigned likelihood to be disordered for a given residue. Predictions from each approach are combined in a weighted average and smoothed to provide a final output. The more recent incarnation **PrDOS-CNF** applies the same methodology using conditional neural fields instead of SVMs (Monastyrskyy et al. 2014). The same group also maintains a meta-predictor, **metaprdos2**, which combines the predictions of five prediction methods including DISOPRED2 and POODLE-S (see below) in an SVM classifier (Ishida and Kinoshita 2008; Monastyrskyy et al. 2014).

DISOPRED3 employs three prediction models: a neural network tuned to capture long disordered regions, an SVM, and a nearest neighbor predictor that can be adapted to new experimental data since it does not require any training (Jones and Cozzetto 2015). The output of the three predictors is combined into the final prediction using a simple window-based neural network. A new feature of DISOPRED3 is the prediction of putative protein binding sites within the identified IDRs. This is performed by three separate SVMs with increasingly complex input signals, all of which are based on sequence or evolutionary information.

POODLE is an SVM-based family of disorder predictors. Three predictors are combined in the meta-predictor POODLE-I such that a specific method is trained for a specific length of disordered segment (Hirose et al. 2010). POODLE-S predicts short IDRs based on their distance from the N-terminus using information from PSI-BLAST PSSMs, as well as using physicochemical features as input (Shimizu et al. 2007). POODLE-L predicts long IDRs (Hirose et al. 2007). While predictions are refined in a second level, the first level of POODLE-L predicts IDRs in a 40-residue window using the amino acid composition of the input sequence, as well as physicochemical features. POODLE-W predicts proteins as (mostly) ordered or as IDPs based on a semi-supervised learning scheme that was trained on 608 proteins with known disorder status and on random samples of 30 000 proteins without labels (Shimizu et al. 2007). Finally, POODLE-I combines these predictors by first running POODLE-L and POODLE-W to determine long disordered regions. The predicted IDRs are truncated based on the output of three secondary structure prediction methods. POODLE-S is then applied to the remaining ordered regions and its predictions are compared with 3D structures of similar proteins, if available. If a region is predicted as ordered by POODLE-S but not resolved in the structure, the prediction for this segment is changed to “disorder.”

Table 7.1 Disorder prediction performance.

Method/score	MCC	B_ACC	AUC_ROC
Prdos-CNF	0.529	0.712	0.907
DISOPRED3	0.531	0.698	0.897
POODLE	0.409	0.781	0.875
metaprdos2	0.385	0.778	0.879
Naive	0.282	0.61	N/A

Shown is the performance of the top disorder predictors in the independent evaluation of CASP10 (Monastyrskyy et al. 2014) measured by three different scores: the Matthews correlation coefficient (MCC), balanced accuracy (B_ACC), and the area under the receiver operating characteristic (AUC_ROC, Box 7.3). For every score, the best performing method is highlighted in bold. Performance of a naive predictor, which always considers the first and last residues as disordered, is given as a baseline. Because of its design it only performs binary predictions, and the AUC_ROC cannot be calculated.

Performance Disorder predictions are commonly evaluated on a per-residue basis. CASP10 (described above) contains the most recent assessment for the accuracy of predicting proteins with regular structures using 28 different prediction methods (Monastyrskyy et al. 2014). The best performers, such as POODLE and metaprdos2, reached balanced accuracy (B_ACC) values of 78% with most other methods only slightly worse than a naive baseline method that reaches 61% (Table 7.1). Prdos-CNF and DISOPRED3 were the top performers in terms of AUC_ROC curve (0.9). Among the findings of the CASP10 assessment was that prediction of IDRs that are not at the termini of the protein are harder to predict and that there is no clear correlation between IDR length and the difficulty of their prediction. Generally, performance increased only slightly in the six rounds of CASP in which disorder prediction was assessed.

Predicting Protein Function

The primary reason for our interest in protein structure and its prediction is to learn about protein function. Being able to predict aspects of protein structure and function directly from the protein sequence enables fast functional annotation of the wealth of sequences available and can guide further experimental studies and validation.

Synopsis

Most protein sequences are predicted from genomics data and the only way to study their function is through computational protein function prediction. The most widely used approach for predicting function is by transferring known functional annotation from homologs. However, large-scale assessments of homology transfer show that this approach has caveats and must be applied carefully (Rost 2002; Ashkenazi et al. 2012). Moreover, many proteins do not have annotated homologs; thus, many computational methods for protein function prediction have been developed to use other information, such as sequence features with known functional properties.

There are many aspects of protein function, including the protein's cellular location, interacting molecules, and biological processes that it participates in. Here, we focus mostly on aspects of molecular function, defined as physicochemical processes that the protein molecule can participate in, such as an enzymatic activity, binding other proteins or molecules, or functioning as a transporter. We begin with protein motifs and domains, considered to be the functional units of a protein. Then, we discuss the prediction of Gene Ontology (GO) terms, a dictionary capturing the various aspects of protein function (Gene Ontology Consortium 2000, 2015). A

related endeavor is the Critical Assessment of Function Annotation (CAFA), which, over the last 6 years, has established itself as a major resource for gauging how well protein function prediction works. We also provide several examples and tools for methods that predict subcellular localization, protein interaction binding sites, and the prediction of the effect of single amino acid sequence variants (SAVs) on protein function.

Motifs and Domains

Background Proteins contain many types of short and longer sequence regions that are responsible for carrying out particular functions. Motifs are typically short; for example, the nuclear localization signals responsible for import into the nucleus, typically ranging from 6 to 20 residues, or the six-residue-long motif characterizing the active site of serine proteases. Structural domains, on the other hand, are defined as protein fragments that fold independently into a characteristic structure, exist in many otherwise different proteins, and carry out some functions. Most such structural domains are 100–500 residues long (Liu and Rost 2003) and are the basis for the modular architecture of proteins. For example, the Pleckstrin homology domain plays a role in targeting of proteins to membranes; it is found in both dynamin, which is involved in vesicle processing, as well as many kinases such as those of the Akt/Rac family that play a role in signal transduction. Finding a motif or domain in a newly discovered sequence can offer insights into its structure and function.

Motifs and domains are typically originally discovered by identifying conserved regions in a set of proteins, assuming that more conserved protein sequence regions are more important for function than less conserved regions. These are then functionally characterized using experiments, such as mutating the motif and observing what function the protein loses. Many databases are dedicated to cataloguing such information from a variety of sources. Motifs and domains are described by sequence patterns learned from known examples. For example, a simple pattern consisting of four amino acids, where the first is always an alanine, followed by any amino acid, then either glutamic or aspartic acid, followed by a final histidine, can be captured as a regular expression. However, most sequence patterns are larger and more complex – for instance, some amino acids may appear more often than others in a set of motifs. Such patterns are better expressed using PSSMs or HMMs (Box 7.1), constructed based on an MSA (Chapters 3 and 8) using tools such as HHsearch and HMMER (Söding 2005; Eddy 2011). These methods are limited to analyzing known patterns, and prediction methods have been developed that predict domains, domain boundaries, or domain linker regions from sequence alone, using amino acid propensities or homology information from sequence alignments. Other methods employ structural templates, that is, they compare the query protein with known protein structures and their domain annotations.

Databases **InterPro** is the primary resource that catalogs information about sequence motifs and domains (Finn et al. 2016) (Figure 7.7a). It collects data from 14 member databases, each with its own specialization, and makes the annotations available through a unified search tool, InterProScan, which exists online as well as a standalone tool (Jones et al. 2014). InterPro comprises the following databases.

- **PROSITE**, which contains regular expression-like patterns, as well as profiles which identify protein families, domains, and functional sites (Sigrist et al. 2013).
- **Pfam**, a database of profile HMMs, which are built from manually curated seed alignments that are specific to a protein family (Finn et al. 2014a, 2016).
- **SMART**, which contains manually curated HMMs that identify domains and can provide information on orthologs based on its annotation process (Letunic et al. 2015).

- **TIGRFAM**, containing families represented by curated HMMs that have been collected with an emphasis on longer annotations that describe more specific functions (Haft et al. 2013).
- **SUPERFAMILY**, a collection of HMMs derived by SCOP from 3D structures, modeling all regions similar to known 3D structures (Oates et al. 2015).
- **PRINTS**, which, as the name suggests, is a set of protein family “fingerprints” that is manually curated from MSA (Attwood et al. 2012).
- **PRODOM**, a set of automatically generated domain families based mostly on PSI-BLAST searches from SCOP domains (Bru et al. 2005).
- **CATH-Gene3D**, which contains profile HMMs based on superfamily annotations in CATH (see following paragraph) (Lam et al. 2016).
- **PIRSF**, which clusters sequences into homeomorphic families whose members have similar sequences over their entire length and whose protein domains are in the same order (Wu et al. 2004).
- **PANTHER**, containing families of homologous genes, each with a phylogenetic tree that displays the evolutionary relationship between the family members (Mi et al. 2016).
- **HAMAP**, providing general protein family profiles built from manually curated MSAs based on sequences that are synchronized with those in UniProtKB (Pedruzzi et al. 2013, 2015).
- **SFLD**, one of the most recent additions to InterPro (Akiva et al. 2014) that contains manually curated hierarchical clusters of enzymes and aims to provide an alternative to purely sequence- or homology-based clustering that cannot accurately account for the evolution of these proteins.
- **CDD**, a meta-database that incorporates domain definitions from Pfam, SMART, clusters of orthologous group (COG), PRK (Klimke et al. 2009), and TIGRFAM (Finn et al. 2016); also contains domain boundaries determined from known 3D structures.
- **MobiDB** contains consensus predictions of disordered regions longer than 20 residues (Necci et al. 2017).

These descriptions reveal the inherent (and intended) redundancy of InterPro. The reason for this redundancy lies in the hope that alternative approaches to the same problem can lead to a more complete and reliable annotation of motifs and domains. Furthermore, some databases can provide more specific annotations than others, allowing the user to choose the desired level of granularity from the combined results.

Two other important related databases are the above-mentioned SCOP and CATH. **SCOP2** contains manually curated domains from PDB structures organized in a hierarchical format, defined as a directed acyclic graph (Andreeva et al. 2014). **CATH** contains semi-automated structure-based domain annotations at four hierarchical levels (Sillitoe et al. 2015). To offer more specific descriptions, an extension of CATH, called **SOLID**, clusters superfamily members at increasingly higher sequence identity cut-offs into another five levels (Greene et al. 2007). CATH recently added **FunFams**, which are represented by HMMs that have been created by clustering sequences based on specificity-determining positions (SDPs) (Das et al. 2015). SDPs are positions that are unique to a protein subfamily as compared with the overall family, and thus presumably are important for function specific to the subfamily. SDPs are a more sensitive pattern determinant than just conserved sequence positions. Finally, **COG** is a database containing COGs which are formed from full-length proteins from microbial genomes (Galperin et al. 2015).

No single database captures every aspect of function annotation. Therefore, meta-databases such as InterPro are crucial resources for proteins without annotations since they allow researchers a straightforward comparison and to judge reliability of the information presented while maintaining transparency of their origin. Another approach has been pursued by **Swiss-Prot** (Bairoch and Boeckmann 1994; Boutet et al. 2016), a database that has largely

(a)

(b)

Figure 7.7 Annotations of human tumor suppressor P53 (P53_HUMAN). (a) InterPro (Finn et al. 2016) shows the entry as belonging to one single family, namely IPR002117. Families are sets of proteins assumed to share the evolutionary origin and inferred to be similar in function and structure based on homology. Domains (distinct functional or structural units) are shown, followed by a list of the specific hits against all InterPro databases (see text for details). Here, the identifiers of each segment in the source database are shown on the right and as a mouse-over. For example, PF08563 is a reference to the Pfam family “P53 transactivation motif” and cd08367 links to the CDD entry for “P53 DNA-binding domain.” (b) neXtProt expertly curates the current knowledge about protein function (Gaudet et al. 2017). While many annotations originate from UniProtKB/Swiss-Prot, neXtProt carefully adds information based on experimental data.

been assembled and curated by experts who read the literature and use many of the databases and methods described in this chapter. Over recent years, Swiss-Prot, which is part of UniProtKB (UniProt Consortium 2016; see Chapter 1), has moved its objective from “many annotations” to “deep annotations,” i.e. from the effort to annotate as many proteins as possible to that of focusing on providing as detailed annotations as possible for fewer proteins. **neXtProt** is a related effort that focuses solely on collecting the most detailed possible annotation of all human proteins (Gaudet et al. 2017) (Figure 7.7b).

Methods Motif- and domain-finding methods used by the databases and tools mentioned above essentially are derivatives of sequence alignment methods, such as those discussed in Chapter 3. In the following section, we cover methods for the identification of structural domains using only information about protein sequence.

DomCut is one of the earliest and simplest approaches for the prediction of domain linker regions (Suyama and Ohara 2003). At every position over a sliding window of size 15, it compares the amino acid frequency between domain and linker regions, as calibrated from a high-resolution set of structures from PDB.

Scooby-Domain identifies domains by their characteristic hydrophobic amino acid composition. The method calculates the fraction of hydrophobic residues in sliding windows of all possible sizes, from the smallest to the largest observed domain (Pang et al. 2008). This leads to a matrix that contains the average hydrophobicity for every combination of window size and central residue in the window. Starting from the 10 best-scoring values in the matrix, the A* heuristic search algorithm is used to filter through a set of alternative domain architectures. The method was further improved by integration of domain boundary predictions by DomCut and PDLI (Dong et al. 2006), as well as homology information. As a special feature, Scooby-Domain can also predict a domain composed of segments that are not continuous in sequence. The output of the network is first smoothed and domains are then assigned using a pattern-matching algorithm that merges two regions with high domain scores.

DOMpro predicts protein domains with a 1D recursive neural network that is fed the whole sequence as input, with 25 inputs per protein residue (Cheng et al. 2006). Twenty of these input units code for amino acid probabilities from a PSSM, capturing evolutionary information, while the other three code for predicted secondary structure and two for solvent accessibility features at that position.

Dom-Pred is a domain prediction web server consisting of two components (Bryson et al. 2007): DomSSEA recognizes known domains by predicting the secondary structure of a query sequence, then comparing the secondary structure profile against that of domains deposited in PDB. DPS, on the other hand, tries to predict domain boundaries of yet-uncategorized domains. This exploits the fact that local alignments resulting from a PSI-BLAST search have been observed to create consistent N- and C-terminal signals. These boundaries likely mark domains that are expected to be conserved.

Performance Predictions of domains can be assessed on two levels. The first is to correctly predict the number of domains in a protein. For this measure, Dom-Pred reports a performance of 85% correctly identified domains using homology inference (DomSSEA) and 68% from sequence alone (DPS). DOMpro reports 69% correctly identified domains. The second and more difficult level involves the correct identification of the domain boundaries – or, alternatively, the linker regions between domains. Here, DomCut reports a sensitivity of 53%, similar to Scooby-Domain and DPS both with 50%. DomSSEA improves the performance to 55%. Scooby-Domain also claims a precision of 29%, compared with 85% (DomSSEA) and 72% (DPS). Finally, DOMpro reaches an accuracy of 25% for proteins with two domains and 20% for more than two. Unfortunately, these performance measures can only be compared to a limited degree, as they were calculated on different sets of data and using varying definitions of how

much a predicted boundary can deviate from the experimental annotation while still counting as correct.

CASP8 provided the last independent evaluation of domain boundary predictions (Ezkuordia et al. 2009). Eighteen predictors were assessed on up to 122 targets with manually curated domain definitions. The best methods predicted the correct number of domains in up to 90% of the cases and also performed well in determining exact domain boundaries (as assessed by how close predictions are to the experimental annotation). All of the top-scoring methods made use of structure templates where available, and the CASP assessors noted that performance significantly decreased for targets which had to be predicted *ab initio*, without prior structural knowledge. Among the methods mentioned above, only Dom-Pred was evaluated during CASP8, predicting the correct number of domains for less than 70% of the targets. It should also be kept in mind that single-domain proteins are over-represented among all proteins of known 3D structure (i.e. all CASP targets) and that predictions involving these single-domain proteins yield significantly higher performance (Liu and Rost 2004; Ezkuordia et al. 2009).

Gene Function Prediction Based on the Gene Ontology

Background To systematically predict and evaluate protein function requires a well-defined dictionary of terms describing such functions. GO (Gene Ontology Consortium 2000, 2015) is the standard and most comprehensive such dictionary. GO captures multiple aspects of protein function, described using three different hierarchies/ontologies: one for molecular function (MFO; e.g. “alcohol dehydrogenase activity”), one for biological process (BPO; negative regulation of eye pigmentation), and one for cellular component (CCO, “extracellular matrix component”). Each of these consists of terms ordered hierarchically in a directed acyclic graph, with the most specific descriptions of function at the graph’s leaves. For example, parents of “negative regulation in eye pigmentation” in BPO include “regulation of pigmentation during development” and, simply, “pigmentation.” GO, in collaboration with genome and protein database groups, also maintains annotations of these terms to genes for multiple species, with each annotation associated with an evidence code that provides information about where the annotation information is derived from (e.g. “inferred from sequence similarity” or “traceable author statement”).

Methods **Metastudent** (Hamp et al. 2013) predicts GO terms for a protein based on homology. BLAST (Chapter 3) is used to find similar sequences with known GO annotations, and these annotations are then transferred to the query protein. If no homolog is found, no prediction is made. In general, this method has relatively low reliability, but certain terms can be highly reliably predicted.

COGIC uses annotation transfer by homology and machine learning to predict GO terms (Cozzetto et al. 2013). Specifically it uses: (i) annotation transfer from PSI-BLAST hits against UniRef90 with at least 85% alignment coverage; (ii) a naive Bayes classifier based on text-mining data from UniProtKB/Swiss-Prot; (iii) a naive Bayes classifier assessing the association of the frequency of amino acid 3-mers with specific GO terms; (iv) predictions by the SVM-based method FFPred for eukaryotic targets (described below); (v) further annotation transfer based on distant relationships described by the orthologous groups found within the EggNOG database (Huerta-Cepas et al. 2016); (vi) a simple neural network, based on a similarity score that compares the PSSMs of query sequences with those from the annotated dataset; and (vii) GO term predictions based on high-throughput data provided by the method FunctionSpace for human proteins (Lobley 2010). Predictions by these seven components are then combined and propagated to protein annotations based on the GO graph structure.

FFPred 3.0 is an SVM-based *de novo* function predictor focusing on human proteins (Cozzetto et al. 2016). Therefore, it is a particularly helpful tool when annotation transfer

cannot be applied or when annotation transfer methods provide few results. As the prediction of GO terms is a large multi-label problem, a single protein can be associated with many GO terms, but SVMs are binary classifiers; FFPred trains a total of 868 separate SVMs, each predicting a single GO term. The input feature set for the SVMs consists of sequence features including amino acid composition, predicted secondary structure, transmembrane segments, disordered regions, signal peptides, subcellular localization, and more.

FunFams, short for Functional Families, are clusters of protein domains which are likely to share the same function (Das et al. 2015). To construct FunFams, all sequences from a CATH superfamily with a sequence identity of 90% are clustered (Sillitoe et al. 2013, 2015). These initial clusters are then further merged using profiles created from the MSAs of each cluster in an iterative fashion until a single cluster remains. In the process, a hierarchical clustering tree is constructed. Next, the optimal cut of the tree is determined such that the remaining parent nodes represent FunFams, considering SDPs (described in Motifs and Domains, Databases). The resulting FunFams can then be used to predict GO terms. All superfamilies identified in a protein are linked to all corresponding FunFams and all the associated GO terms and their GO DAG ancestors are transferred to the query protein.

Performance Because of the hierarchical structure of GO, evaluation metrics consider not just perfect matches of predicted GO terms but also their parents. Furthermore, an evaluation can be either protein centric, measuring how well all GO terms of a single protein have been predicted and averaging over all proteins, or term centric, evaluating how well a specific term has been predicted over all proteins in the set.

The most commonly reported score by method developers is the protein-centric F_{\max} , corresponding to the maximal F1 score (see Box 5.4) achieved along the precision-recall or the ROC curve (Box 7.3). This is also the main score in the CAFA algorithms (Radivojac et al. 2013), the de facto standard in evaluating GO term prediction methods.

CAFA is the equivalent of CASP for the protein function prediction community. The idea is to define a set of target proteins for which all participants must submit their predictions by a given deadline. This is followed by an annotation growth phase of around 8 months during which the organizers wait for the accumulation of experimental data to be annotated by GO curators; these data can then be used to evaluate the previously submitted prediction results. In the first CAFA experiment conducted from 2010 to 2011, only two GO aspects were evaluated: MFO and BPO. CAFA2 (2013–2014) extended this evaluation by also adding the third GO ontology – CCO – and further added terms describing human phenotypic abnormalities as cataloged in the human phenotype ontology (HPO; Köhler et al. 2016). CAFA3 started in 2016 and finished evaluations in 2019. It expanded the scope by including predictions of metal- and nucleotide-binding residues, as well as a manually curated set of moonlighting proteins that perform multiple distinct functions, only one (or none) of which is made known to participants in the challenge.

Results for the protein-centric F_{\max} from CAFA2 are shown in Table 7.2 (Jiang et al. 2016). Compared with the PSI-BLAST baseline, overall performance is best for MFO, worse for BPO, and as low as a naive prediction using CCO. However, the latter low performance was likely due to a problem with the assessment design, as a few very common terms such as “organelle” seem to have dominated the evaluation. When adapting the alternative scoring scheme (S_{\min}) that de-emphasizes general terms over more descriptive ones further down the GO graph, the naive prediction performance dropped significantly – below that of the 10 best prediction methods. Using S_{\min} for scoring, FunFams (MFO and BPO) and jfpred (CCO, based on COGIC/FFPred) join the ranks of the best methods. Independent of weighting, no method was able to perform better than the naive baseline on human phenotypes described by HPO. The generally bad performance could be attributed to the fact that target proteins on average have many more HPO terms annotated than they do for the three GO ontologies. Furthermore, HPO is limited to human, while GO allows the transfer of annotations from homologs in other organisms.

Table 7.2 Performance of selected gene ontology term prediction methods in CAFA2.

F_{\max}			
Prediction method	Molecular function (MFO)	Biological process (BPO)	Cellular component (CCO)
jfpred (FFPred+COGIC)	0.544	0.446	0.35
FunFams	0.563	0.43	0.352
Metastudent	0.523	0.441	0.288
Best method	0.595	0.468	0.372
PSI-BLAST Baseline	0.451	0.347	0.251

Shown is the performance of the Gene Ontology (GO) term prediction methods evaluated in the most recent independent assessment performed by CAFA2 (Jiang et al. 2016), based on the primary score, F_{\max} , for the three GO ontologies. For reference, *Best method* shows the best performance achieved in the respective ontology for any method. All values should only be seen as a rough performance estimate, as the coverage of methods shown here differs and error estimations have been omitted. The original publication contains additional performance scores that are complementary to F_{\max} .

Overall, performance has improved from CAFA1 to CAFA2 (Jia and Liu 2006). It is furthermore encouraging that the best methods do not perform significantly worse for difficult targets that have low sequence similarity to the training data. Finally, while several methods stand out for one or two measures, there is no single method that consistently performs best across all ontologies and scoring schemes.

Subcellular Localization

Background Predicting the subcellular localization of proteins computationally is an important challenge in bioinformatics, as the compartment in which a protein functions natively is informative about its function. Experimental studies have shown that proteins may travel between different subcellular compartments, yet most of them are functional within a single compartment for most of their lifetime (Huh et al. 2003; Foster et al. 2006). Furthermore, the cellular sorting mechanism that directs proteins to their main location of function is relatively well understood, providing useful features for computational prediction methods, and experimental localization data are available in public databases for many proteins, providing useful training and test data. The manually annotated database UniProtKB/Swiss-Prot (O'Donovan et al. 2002) contains experimental localization information for more than 30 000 proteins (release 2018_08). However, these constitute only 0.03% of all known proteins in the current UniProt release (UniProt Consortium 2016). Thus, the vast majority of proteins do not have any experimentally determined cell location information, so computational prediction methods can help fill this gap.

The best computational methods achieve impressive levels of prediction performance (Gardy and Brinkman 2006; Horton et al. 2007; Hu et al. 2009; Goldberg et al. 2014) and are routinely used for large-scale protein annotation (Graessel et al. 2015; Ramilowski et al. 2015). However, most of these methods focus on one or a few cellular compartments or specific organisms. The most reliable localization predictions result from careful homology-based inference, i.e. where localization information is transferred from an experimentally annotated protein to its unannotated sequence homolog. Unfortunately, this method cannot be applied to most proteins because of limited existing cellular location information. For these, *de novo* machine learning methods provide fairly reliable results. Other automatic methods annotate proteins by mining biological literature and molecular biology databases (Nair and Rost 2002). For example, a simple approach could be to compare annotated UniProtKB keywords between the query protein and proteins of known subcellular localization. Location annotations are

transferred from a protein with similar keywords. Finally, meta-predictors integrate different methods, using the most reliable prediction for the final prediction.

Methods **LocTree3** is the latest generation of the LocTree method, employing a hierarchical structure of SVMs to mimic the protein-trafficking pathway in a cell and predict protein localization (Goldberg et al. 2014) (Figure 7.8). LocTree3 predicts 18 subcellular localization classes in eukaryotes, six in bacteria, and three in archaea. LocTree3 is a hybrid approach that first uses a PSI-BLAST search to transfer any known localization information to a query protein from its sequence homolog. If no homology-based information is available, it uses a de novo-based method called LocTree2 that employs a profile-kernel SVM, using an evolutionary profile-based conservation scores of short stretches of k consecutive residues (k -mers) as input (Goldberg et al. 2012).

MultiLoc2 predicts protein localization by integrating the overall amino acid composition with known cellular sorting signals, phylogenetic profiles, and GO (Gene Ontology Consortium 2015) terms for the prediction of protein localization in eukaryotes (Blum et al. 2009). MultiLoc2 is available in two versions: MultiLoc2-LowRes can be applied to eukaryotic globular proteins and predicts five localization classes, while MultiLoc2-HighRes additionally covers transmembrane proteins and predicts 11 eukaryotic subcellular localization classes. MultiLoc2 uses a two-level prediction approach, where the first layer consists of several SVMs processing different encodings of the query sequence, forwarding their output to the second-level SVM-based classifiers that produce probability estimates for each localization class.

DeepLoc predicts 10 different eukaryotic localization classes (Almagro Armenteros et al. 2017). Unlike the methods mentioned above, DeepLoc performs its prediction ab initio using only the sequence and no ancillary homology information. The method first uses a convolutional neural network to extract sequence motifs of varying lengths that are then used as

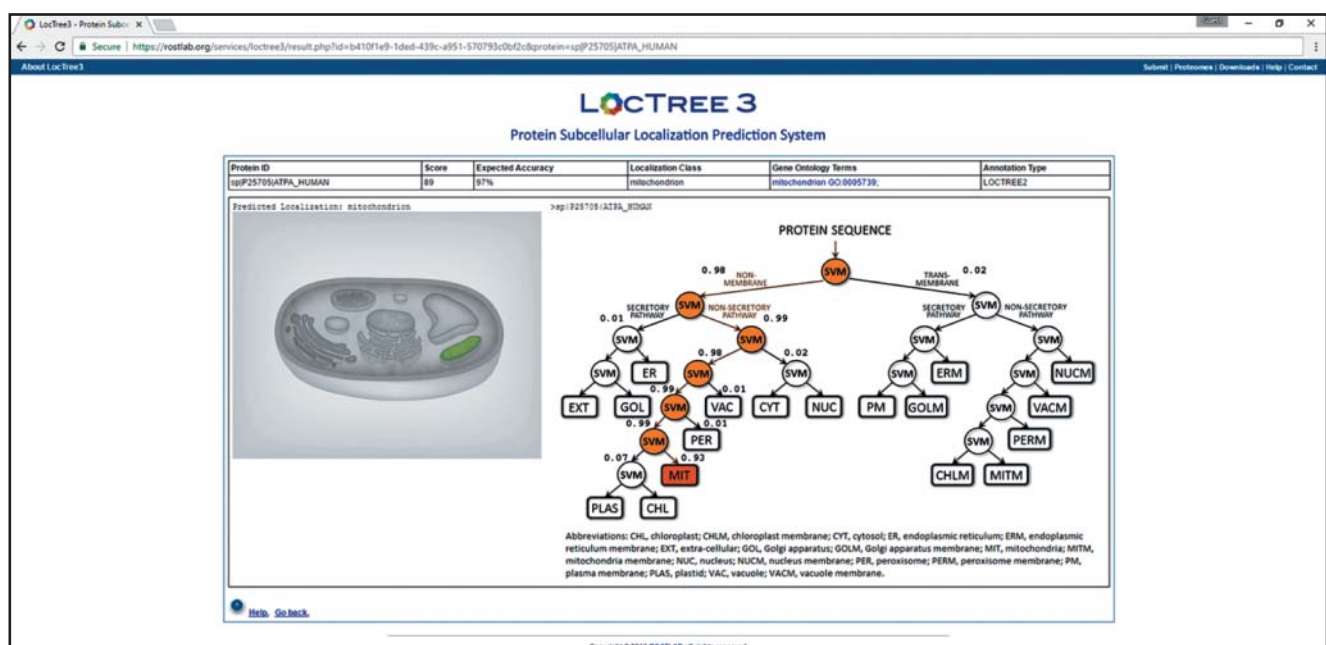


Figure 7.8 Prediction of subcellular localization. Visual output from LocTree3 (Goldberg et al. 2014), a web server predicting subcellular localization. Upon submission of a sequence (and specification of its kingdom: eukaryota, bacteria, or archaea), the predicted localization, along with the corresponding Gene Ontology terms (if available), are shown. The last column (Annotation Type) denotes whether the result was obtained through a homology search or, as is the case here, predicted by the machine learning model. The tree on the right is a visualization of the hierarchical network of support vector machines (SVMs). Highlighted in orange is the path that was used to predict the example, the alpha subunit of human ATPase, as being localized to the mitochondrion.

input for a recurrent neural network and fed through additional filtering layers. The final step performs predictions using a hierarchical approach similar to that of LocTree3.

Performance Generally, the performance of subcellular localization prediction methods can be assessed using a binary accuracy score (Box 7.3; see also Box 5.4), even though many methods predict 10 or more classes. In general, newer methods that integrate more sources of information outperform older and simpler methods (Hu et al. 2009; Mooney et al. 2013; Goldberg et al. 2014). The reported performance levels suggest accuracies >85% for extracellular proteins, ~80% for those found within the plasma membrane and nucleus, >65% for cytoplasmic proteins, and >70% for mitochondrial proteins. DeepLoc, the most recent method, reports an overall accuracy of 78% based on their newly assembled dataset. Using the same dataset, LocTree2 reaches 61% accuracy, while MultiLoc2 reaches 56% accuracy.

Protein Interaction Sites

Background Most proteins do not function alone. Instead, they work with other molecules within molecular complexes (Rao et al. 2014; Keskin et al. 2016) to perform specific functions. Protein interactions and resulting interaction networks are covered in greater depth in Chapter 13. Here, we focus on describing methods that predict physical, non-covalently bonded protein–protein interaction sites for an individual protein 1D sequence. Many methods are also available to predict protein–protein interaction sites based on known 3D structures, and the reader is referred to two comprehensive references for more information on these methods (Esmailbeiki et al. 2016; Keskin et al. 2016).

There are many types of protein interaction binding sites. For instance, a site could be defined by a large surface of a protein's 3D structure, corresponding to amino acid positions scattered across a 1D sequence that are close in the 3D fold, which is used to bind to other large proteins. Small sites that bind small molecules are prevalent in enzymes, and usually define the enzyme active site. Finally, short linear sites are continuous amino acid stretches, often found in disordered regions, that bind proteins (Tompka et al. 2014). Protein interactions are usually mediated by domains and motifs, each with their own characteristic binding sites and preferred binding patterns in their partner molecules (Pawson and Nash 2003). Proteins may have multiple interacting partners and binding sites and some sites may bind to more than one partner, which may lead to competition among the partners. Interestingly, even between two identical proteins A and B, we surprisingly observe alternative interfaces that are biologically meaningful (Hamp and Rost 2012).

Typically, binding interfaces, which can involve 20–50 residues, are defined by having a large difference in ASA between the monomer and the complex *or* by the distance between two residues within a specific protein 3D structure (Figure 7.3). However, the characterization of binding interfaces is not trivial, so a specific set of tools has been developed that enable the user to discern artifacts from true interactions (Tuncbag et al. 2009; Keskin et al. 2016). Binding sites can also be experimentally identified using a range of methods, such as alanine scanning, whereby each position in a protein sequence is mutated to alanine one by one, and the resulting change in binding strength is then measured. Positions that affect the binding strength when mutated are part of the binding site. Some interacting residues contribute more to binding than others. Those contributing most are often referred to as *hot spots* (Tuncbag et al. 2009; Morrow and Zhang 2012), with about 10% of all binding residues being hot spots. These residues are often buried and differ by their native amino acid composition, with tryptophan, arginine, and tyrosine being particularly common. As hot spots are prime drug targets, several tools have been developed to predict them (Keskin et al. 2016). Newer methods for binding site prediction involve searching for patterns of correlated mutations present in MSAs, based on the concept that evolutionary mutations in one site must be compensated for in a partner site for

the interaction to be maintained (Marks et al. 2012). Several tools have been developed recently based on these advances, such as FILM3, EVfold, and EVcomplex (Marks et al. 2011; Nugent and Jones 2012; Hopf et al. 2014). Evaluated as part of the CASP11 and CASP12 assessments, there have been tremendous advances in the development of these methods, and it is expected that they will be further improved in the near future (Kinch et al. 2016; Yang et al. 2016b; Wang et al. 2017; Schaarschmidt et al. 2018).

About 5% of all eukaryotic proteins are assumed to bind nucleotides; their functions include processes such as the regulation of gene expression, DNA replication, and DNA repair (Yan et al. 2016). To predict binding residues for these cases, a separate set of specialized predictors has been developed that are based on either the sequence or template 3D structures where available (Zhao et al. 2013).

Many databases collect protein interaction sites (Tuncbag et al. 2009; de Las Rivas and Fontanillo 2010; Keskin et al. 2016). Protein binding interfaces and their interacting residues are stored in PISA (Krissinel and Henrick 2007), Inferred Biomolecular Interactions Server (IBIS) (Shoemaker et al. 2012), IPfam (Finn et al. 2014b), PIFACE (Cukuroglu et al. 2014), and 3did (Mosca et al. 2014). Results from experimental alanine scanning mutagenesis are available within BID (Fischer et al. 2003), as well as from a legacy database called ASEdb (Thorn and Bogan 2001).

Methods Predicting PPI interface residues. A popular approach for predicting binding sites in proteins is to learn their physicochemical patterns from known examples and then search for these patterns in other protein sequences (Ofra and Rost 2003a,b; Esmailbeiki et al. 2016). Ofra and Rost (2003a,b) developed a prediction method based on a neural network with a sliding window of size nine and using only the sequence as input. Šikić et al. (2009) more recently showed that this remains a viable approach. Subsequent methods improved performance by including evolutionary information, as well as predicted secondary structure and solvent accessibility data (Res et al. 2005; Wang et al. 2006; Ofra and Rost 2007; Chen and Jeong 2009; Chen and Li 2010). Developers of the **PSIVER** method found that predicted accessibility information is also highly informative, even if used alone (Murakami and Mizuguchi 2010). **HomPPI** is a purely homology-based predictor that transfers binding site information from known sites in homologous proteins that are part of molecular complex 3D structures in PDB (Xue et al. 2011). ELM is a database of short linear protein binding site patterns, typically described by regular expressions, with associated tools for identifying known patterns in a query sequence (Gouw et al. 2018).

Predicting protein–DNA and –RNA binding. **DBS-PSSM** extends an earlier approach for the prediction of DNA-binding residues by training a relatively small neural network (103 nodes in total) with a sliding window of five and conservation information from PSI-BLAST PSSMs (Ahmad and Sarai 2005). **DP-Bind** is a consensus approach combining the prediction results from an SVM and two logistic regression models (Hwang et al. 2007; Rezáková et al. 2008). Both of these methods found that performance is significantly increased when using homology information. **SomeNA** (Hönigschmid 2012; Yachdav et al. 2014) uses a hierarchical set of neural networks that predict whether a protein binds DNA or RNA and, if so, where it binds. RNA-binding residues are also predicted by **Pprint** using an SVM with PSSM input (Kumar et al. 2008). **RNABindRPlus** tackles the same task using the same input to an SVM and a window size of 21 (Walia et al. 2014). A logistic regression classifier is used to combine the predictions with the results from the homology-based predictor **HomPRIP**, which generally achieves higher performance but cannot be applied to query proteins without homologs.

Performance Predicting PPI interface residues. Predicted protein–protein interaction (PPI) sites are typically evaluated using the same set of standard scores as outlined for other binary classification tasks (see Box 5.4). Without a recent independent evaluation, the only

values available are those provided by method authors. These values have been determined based on different test data and are therefore not comparable. They are also highly variable. Independent of these difficulties, performance has been reported to have reached a plateau (Esmailbeiki et al. 2016).

Predicting protein–DNA and –RNA binding. A recent review by Yan et al. (2016) has comprehensively studied 30 prediction methods of DNA- and RNA-binding residues, evaluating nine on an independent dataset of 3D structures. The authors found that most predictors reach a similar performance for the prediction of DNA-binding residues with an AUC_ROC around 0.79 (Box 7.3). These results are encouraging, but work remains to be done and predictors do differ in terms of whether they emphasize higher specificity (most) or sensitivity. In fact, the area under the ROC curve may not be the most appropriate way to evaluate such prediction methods. For example, DBS-PSSM has the best sensitivity, at 72%, but a specificity of only 75%, significantly lower than other methods that reach more than 90%. A consensus predictor developed by Yan et al. (2016) that combined prediction results using a logistic regression approach outperformed all individual predictors. Similar results describe the prediction of RNA-binding residues, albeit with lower AUC_ROCs (0.724 for RNABindR and 0.681 for Pprint). Interestingly, these methods seem to capture general properties of nucleic acid binding. For example, an RNA-binding site prediction method also predicts many DNA-binding residues as positives. To counter this, new methods, such as SomeNA, that train on both data classes (DNA and RNA) in an attempt to distinguish them have been proposed (Yachdav et al. 2014). One important problem that has been benchmarked poorly so far is the degree to which a method confuses proteins that bind nucleotides and those that do not. All the above performance estimates apply if and only if the protein is already known to bind DNA or RNA.

Effect of Sequence Variants

Background Any two unrelated individuals differ in their genomes at around 5 million sites (Auton et al. 2015) and by about 20 000 SAVs. An important question is whether these mutations affect protein function. Because of the large number of possible protein mutations, it is currently impossible to evaluate all of them using experimental methods, so computational prediction methods are needed to fill the gap. Using such methods, surprising findings are revealed. For instance, many SAVs in healthy individuals appear to have strong impact upon function, common SAVs appear to affect function more than rare SAVs, and SAVs between individuals have, on average, more impact than those seen between human and related organisms (such as apes or mice; Mahlich et al. 2017). Most methods, including the ones presented below, are either limited or strongly biased toward human sequence variants. Chapter 17 provides more information about how these methods are used in practice.

Methods **SIFT** was one of the first methods to predict SAV effects on protein function (Ng and Henikoff 2003; Kumar et al. 2009). The central information used in this method (and most other methods) is to evaluate the SAV in terms of evolutionary sequence conservation identified in an MSA. If the SAV changes the native amino acid to one that is observed in the family of the varied protein, the variant is predicted to be neutral. In contrast, if the SAV alters a conserved position, a change in function is predicted. SIFT uses PSI-BLAST to identify similar proteins and build an MSA, then uses this MSA to predict whether the SAV is likely to be deleterious (i.e. by mutating a relatively conserved position).

PROVEAN is based on the same idea as SIFT but contains some important extensions, including the ability to predict the effect of insertions and deletions and multiple simultaneous variants on protein function (Choi et al. 2012). PSI-BLAST is used to identify sequence families that are clustered at 80% sequence identity. The 45 clusters most similar to the query sequence are selected. For each cluster, the query sequence is compared with the cluster sequences to

compute a delta score using the substitution scores in the BLOSUM62 matrix (see Chapter 3). The arithmetic averages over all scores within each cluster are again averaged to determine the final prediction score. If the score is below an empirically defined threshold, the SAV is considered deleterious. Calculating averages for every cluster before merging results instead of just averaging all sequence scores is important, as large groups of similar proteins might otherwise bias the results; the clustering step reduces redundancy in the data, thereby avoiding this bias. The default score threshold was estimated based on a set of human disease-causing SAVs and common variants (assumed to be neutral) from the UniProtKB (humsavar) database that was then extended to cover insertions and deletions.

PolyPhen-2 predicts SAV effects based on conservation and experimental or predicted information about protein structure (Adzhubei et al. 2010). From a larger set of features, eight sequence-based and three structure-based features were combined within a naive Bayes classifier to determine a final prediction score. PolyPhen-2 was trained on disease-causing mutations annotated in the UniProtKB database. The method aims to predict the effect upon the “system” (a disease) rather than upon the protein (a molecular effect). PolyPhen-2 offers two models: one focuses on SAVs with strong effects that cause a Mendelian disease or disorder involving a single gene, while the other is trained to identify less deleterious alleles that may contribute to complex disease involving more than one gene.

SNAP2 predicts the effect of SAVs on protein function using a system of neural networks (Hecht et al. 2015). As for other methods, the most important signal for prediction comes from evolutionary conservation. To further improve performance, SNAP2 includes protein sequence-based predictions as input, including secondary structure, binding site, and disordered region information. Global features such as the protein length and the amino acid composition are also considered. The method was trained on a dataset that includes human disease-causing SAVs, mostly from the Protein Mutant Database that catalogs how variants affect an experimentally measured level of molecular function. Thus, while SNAP2 predicts how an SAV affects molecular function, this may not correspond to a deleterious effect on the phenotype being studied. SNAP2 was developed to support the idea of “comprehensive in silico mutagenesis,” which attempts to learn the functional effects of all possible protein mutations; this information, in turn, can be used to compute a heat map illustrating the functional effects of each mutation (Figure 7.9; Hecht et al. 2013). Although conservation provides the dominant signal, the additional features under consideration are also important, particularly for proteins with few known similar sequences. Recent analyses show that SNAP2 results are very different for many variants when compared with the predictions from the other methods described here (Reeb et al. 2016).

CADD (Kircher et al. 2014) is the only method presented here that predicts the effect of a variant on genomic sequences rather than for proteins; however, it also can handle SAVs. It can score any SNV or small indel in either coding or non-coding regions. CADD uses a set of 63 features that are combined within an SVM classifier. These features include evolutionary, regulatory, and transcript information (such as transcription factor binding sites or intron and exon boundaries), as well as predicted effect scores from PolyPhen and SIFT. The training was performed on a set of variants that capture differences between human and an inferred human–chimpanzee common ancestor. All of these variants, which also appear in large numbers in the human population, are considered neutral variants since they have been subjected to natural selection. Another set of simulated in silico variants are considered deleterious because they are not under evolutionary constraint. CADD provides a single score measuring the deleteriousness of a variant, independent of what type the variant is and where in the genome it lies. This makes the method more generally applicable than the others described above.

Performance Generally, variant effect prediction is evaluated with the same measures used for other classification tasks such as AUC_ROC or accuracy (Box 7.3). For example, in a recent independent evaluation SIFT, PolyPhen, and CADD reached AUC_ROC values of 0.59–0.63

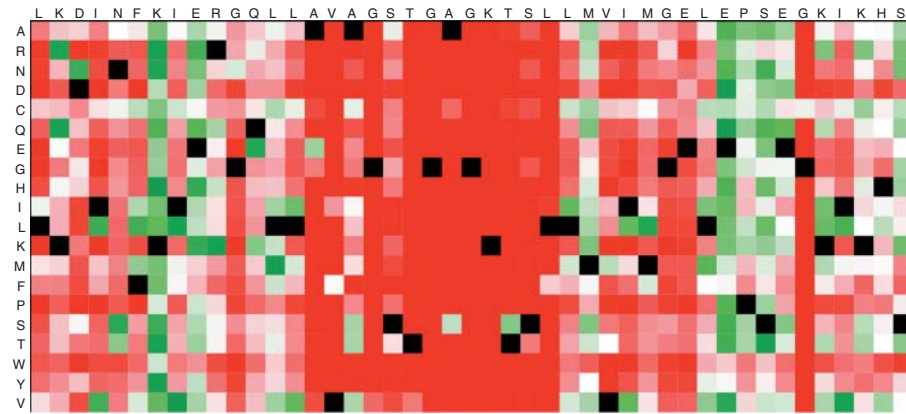


Figure 7.9 From predicting single amino acid sequence variant (SAV) effects to landscapes of susceptibility to change. Shown are the SNAP2 (Hecht et al. 2015) predictions resulting from a complete *in silico* mutagenesis study for the cystic fibrosis transmembrane conductance regulator, as shown by Predict-Protein (Yachdav et al. 2014). Prediction results are represented as a heatmap in which every column corresponds to one residue in the sequence. Rows represent mutations to all other non-native amino acids. Note that not all those SAVs are reachable by a single nucleotide variant (SNV). SAVs predicted as neutral are highlighted in green, while those predicted to affect molecular function are in red. Synonymous mutations are black. While traditionally focusing on a few select variants that are of particular interest, some modern tools are computationally efficient and accurate enough to predict the effect of every possible variant in a protein. While predictors are not sensitive enough to regard every high-scoring variant as potentially interesting, such an approach allows for the identification of sites that may be functionally important, as the effect of every possible kind of variation for that residue is predicted. Here, the clustering of high-effect scores falls exactly into the known nucleotide binding region from residues 458–465. This approach represents one very effective way in which the application of residue-based prediction tools can lead to knowledge at the level of the whole protein.

on a dataset containing variants known to affect protein function. Using a different set of variants that affect the function of transcription factor TP53, performance was significantly higher (0.83–0.87). Performance was even higher on a set of variants implicated in human disease (0.83–0.94). Other published rankings of methods often disagree widely, depending on the datasets used (Thusberg et al. 2011; Dong et al. 2015; Grimm et al. 2015). This high variation exemplifies the difficulties in evaluating the effect of a SAV. One reason for the variation is that the tools described above predict different types of effects, such as the effect on molecular function, a pathway, or the organism in general. One extreme example of the problem is that SIFT, PolyPhen-2, and SNAP2 predictions seem to generally agree on the base set of SAVs with *known* experimental information that should be used as the basis for predictions, but differ substantially on how to apply never-before-seen data such as the natural sequence variation observed between 60 000 individuals (Mahlich et al. 2017) or by random SAVs (Reeb et al. 2016). Another issue at play is ascertainment bias, where well-studied proteins are more likely to be included in training data, leading to overestimation of generalization performance (Grimm et al. 2015). Given these issues and the overall estimated performance of these tools, they are best used to generate hypotheses for further testing (Miosge et al. 2015).

Summary

Seminal discoveries made in the 1960s by Anfinsen and others have clearly established that the sequence of a protein determines its structure and, ultimately, its function. Owing to the relative simplicity with which protein sequences can be obtained experimentally, a large enterprise devoted to predicting structure and function from sequence has emerged. Structure prediction has tremendously matured to the point that some aspects may be considered solved, at least to the extent that current experimental data allow (Hopf et al. 2012). Despite these substantial advances, the general problem of predicting protein function from sequence has not

been solved. The 1D prediction methods presented in this chapter (secondary structure, transmembrane, solvent accessibility, and disorder) are important as input to higher level prediction methods. Fortunately, given the wide range of prediction methods available, many of which are discussed in this chapter, it is possible to annotate protein sequences with a multitude of information, even without any prior knowledge. These predictions will still clearly contain errors – but, once the user understands the strengths and weaknesses of each method, these tools can be incredibly useful toward allowing the user to filter the deluge of sequence data generated today and, hopefully, generate hypotheses that can be experimentally tested. Given that errors may exist in the data that these methods depend upon, it is important to identify the primary evidence used for protein function prediction, whether using best-in-class tools, mapped by high-throughput experiments, or those carefully gathered by experts based on detailed experiments. No existing resource informs users about these situations. Thus, the analysis of the proteins that you are interested in is typically done best by using the best, appropriate prediction tools for any particular question, along with the most reliable database annotations.

Internet Resources

Essential databases and prediction evaluations

CAFA	biofunctionprediction.org/cafa
CAGI	genomeinterpretation.org
CASP	predictioncenter.org
CATH	www.cathdb.info
InterPro	www.ebi.ac.uk/interpro
neXtProt	www.nextprot.org
PDB	www.wwpdb.org
Pfam	pfam.xfam.org
SCOP2	scop2.mrc-lmb.cam.ac.uk
UniProtKB	www.uniprot.org

Prediction of protein structure

BETAWARE	biocomp.unibo.it/savojard/betawarecl
BOCTOPUS2	boctopus.bioinfo.se
PolyPhobius	phobius.sbc.su.se/poly.html
POODLE	cblab.my-pharm.ac.jp/poodle
PrDOS	prdos.hgc.jp/cgi-bin/top.cgi
Proteus	wks80920.ccis.ualberta.ca/proteus
Proteus2	www.proteus2.ca/proteus2
PSIPRED, MEMSAT-SVM, and DISOPRED3	bioinf.cs.ucl.ac.uk/psipred
RaptorX	raptorx.uchicago.edu/StructurePropertyPred/predict
ReProf, TMSEG, and Meta-Disorder	predictprotein.org
SPIDER3	sparks-lab.org/server/SPIDER3
SSpro5, ACCpro5	scratch.proteomics.ics.uci.edu

Prediction of protein function

CADD	cadd.gs.washington.edu
DeepLoc	www.cbs.dtu.dk/services/DeepLoc
DomCut	www.bork.embl-heidelberg.de/~suyama/domcut
DomPred, FFPred 3.0, COGIC	bioinf.cs.ucl.ac.uk/psipred
DOMpro	scratch.proteomics.ics.uci.edu
DP-Bind	lcg.rit.albany.edu/dp-bind

FunFams	www.cathdb.info/search/by_sequence
HomPPI	ailab1.ist.psu.edu/PSHOMPPIv1.3
HomPRIP-NB	ailab1.ist.psu.edu/HomPRIP-NB/index.html
LocTree3	rostlab.org/services/loctree3
MultiLoc2	abi-services.informatik.uni-tuebingen.de/multiloc2/webloc.cgi
PolyPhen-2	genetics.bwh.harvard.edu/pph2
Pprint	crdd.osdd.net/raghava/pprint
PROVEAN	provean.jcvi.org/index.php
PSIVER	mizuguchilab.org/PSIVER
RNABindRPlus	ailab1.ist.psu.edu/RNABindRPlus
ScoobyDomain	www.ibi.vu.nl/programs/scoobywww
SIFT	sift.bii.a-star.edu.sg
SNAP2	rostlab.org/services/snap2web
SomeNA, Metastudent, Ofran, and Rost PPI predictor	www.predictprotein.org

Further Reading

Keskin, O., Tuncbag, N., and Gursoy, A. (2016). Predicting protein-protein interactions from the molecular to the proteome level. *Chem. Rev.* 116: 4884–4909. Keskin et al. give an expansive overview of protein binding in all its facets, covering protein–protein and protein–nucleic acid binding on the protein and residue level, as well as additional topics not covered in this chapter, such as docking and other prediction algorithms based on protein structure instead of sequence.

Moult, J., Fidelis, K., Kryshchuk, A. et al. (2016). Critical assessment of methods of protein structure prediction: progress and new directions in round XI. *Proteins* 84 (Suppl 1): 4–14. This is the most recent evaluation of the CASP experiment, which is an independent assessment of all major aspects of protein structure prediction. Similarly, readers interested in function prediction should investigate CAFA (Jiang et al. 2016) and CAGI (see Internet Resources) for variant effect prediction.

References

- Adzhubei, I.A., Schmidt, S., Peshkin, L. et al. (2010). A method and server for predicting damaging missense mutations. *Nat. Methods.* 7: 248–249.
- Ahmad, S. and Sarai, A. (2005). PSSM-based prediction of DNA binding sites in proteins. *BMC Bioinf.* 6: 33.
- Akiva, E., Brown, S., Almonacid, D.E. et al. (2014). The structure-function linkage database. *Nucleic Acids Res.* 42: 521–530.
- Allis, C.D. and Jenuwein, T. (2016). The molecular hallmarks of epigenetic control. *Nat. Rev. Genet.* 17: 487–500.
- Almagro Armenteros, J.J., Sønderby, C.K., Sønderby, S.K. et al. (2017). DeepLoc: prediction of protein subcellular localization using deep learning. *Bioinformatics.* 33: 3387–3395.
- Aloy, P., Stark, A., Hadley, C., and Russell, R.B. (2003). Predictions without templates: new folds, secondary structure, and contacts in CASP5. *Proteins Struct. Funct. Genet.* 53 (Suppl 6): 436–456.
- Altschul, S.F. and Gish, W. (1996). Local alignment statistics. *Methods Enzymol.* 266: 460–480.
- Andreeva, A., Howorth, D., Chothia, C. et al. (2014). SCOP2 prototype: a new approach to protein structure mining. *Nucleic Acids Res.* 42: 310–314.

- Anfinsen, C.B. (1973). Principles that govern the folding of protein chains. *Science*. 181: 223–230.
- Ashkenazi, S., Snir, R., and Ofra, Y. (2012). Assessing the relationship between conservation of function and conservation of sequence using photosynthetic proteins. *Bioinformatics*. 28: 3203–3210.
- Attwood, T.K., Coletta, A., Muirhead, G. et al. (2012). The PRINTS database: a fine-grained protein sequence annotation and analysis resource-its status in 2012. *Database*. 2012: 1–9.
- Auton, A., Abecasis, G.R., Altshuler, D.M. et al. (2015). A global reference for human genetic variation. *Nature*. 526: 68–74.
- Bairoch, A. and Boeckmann, B. (1994). The SWISS-PROT protein sequence data bank: current status. *Nucleic Acids Res*. 22: 3578–3580.
- Berman, H.M., Westbrook, J., Feng, Z. et al. (2000). The protein data bank. *Nucleic Acids Res*. 28: 235–242.
- Bernhofer, M., Kloppmann, E., Reeb, J., and Rost, B. (2016). TMSEG: novel prediction of transmembrane helices. *Proteins* 84: 1706–1716.
- Blum, T., Briesemeister, S., and Kohlbacher, O. (2009). MultiLoc2: integrating phylogeny and gene ontology terms improves subcellular protein localization prediction. *BMC Bioinf*. 10: 274.
- Boutet, E., Lieberherr, D., Tognolli, M. et al. (2016). UniProtKB/Swiss-Prot, the manually annotated section of the UniProt knowledgeBase: how to use the entry view. *Methods Mol. Biol*. 1374: 23–54.
- Bru, C., Courcelle, E., Carrère, S. et al. (2005). The ProDom database of protein domain families: more emphasis on 3D. *Nucleic Acids Res*. 33: 212–215.
- Bryson, K., Cozzetto, D., and Jones, D.T. (2007). Computer-assisted protein domain boundary prediction using the DomPred server. *Curr. Protein Pept. Sci*. 8: 181–188.
- Buchan, D.W.A., Minneci, F., Nugent, T.C.O. et al. (2013). Scalable web services for the PSIPRED protein analysis workbench. *Nucleic Acids Res*. 41: 349–357.
- Chen, X.W. and Jeong, J.C. (2009). Sequence-based prediction of protein interaction sites with an integrative method. *Bioinformatics*. 25: 585–591.
- Chen, P. and Li, J. (2010). Sequence-based identification of interface residues by an integrative profile combining hydrophobic and evolutionary information. *BMC Bioinf*. 11: 402.
- Chen, C.P., Kernytsky, A., and Rost, B. (2002). Transmembrane helix predictions revisited. *Protein Sci*. 11: 2774–2791.
- Cheng, J., Sweredoski, M.J., and Baldi, P. (2006). DOMpro: protein domain prediction using profiles, secondary structure, relative solvent accessibility, and recursive neural networks. *Data Min. Knowl. Discovery* 13: 1–10.
- Choi, Y., Sims, G.E., Murphy, S. et al. (2012). Predicting the functional effect of amino acid substitutions and indels. *PLoS One* 7 (10): e46688.
- Chou, P.Y. and Fasman, G.D. (1974). Prediction of protein conformation. *Biochemistry*. 13 (2): 222–245.
- Claros, M.G. and Von Heijne, G. (1994). TopPred II: an improved software for membrane protein structure predictions. *Comput. Appl. Biosci*. 10: 685–686.
- Coleman, J.L.J., Ngo, T., and Smith, N.J. (2017). The G protein-coupled receptor N-terminus and receptor signalling: N-tering a new era. *Cell. Signalling*. 33: 1–9.
- Cozzetto, D., Buchan, D.W.A., Bryson, K., and Jones, D.T. (2013). Protein function prediction by massive integration of evolutionary analyses and multiple data sources. *BMC Bioinf*. 14: S1.
- Cozzetto, D., Minneci, F., Currant, H., and Jones, D.T. (2016). FFPred 3: feature-based function prediction for all gene ontology domains. *Sci. Rep*. 6: 31865.
- Crick, F.H. (1958). On protein synthesis. *Symp. Soc. Exp. Biol*. 12: 138–163.
- Cukuroglu, E., Gursoy, A., Nussinov, R., and Keskin, O. (2014). Non-redundant unique interface structures as templates for modeling protein interactions. *PLoS One*. 9: e86738.
- Das, S., Lee, D., Sillitoe, I. et al. (2015). Functional classification of CATH superfamilies: a domain-based approach for protein function annotation. *Bioinformatics*. 31: 3460–3467.
- Deng, X., Gumm, J., Karki, S. et al. (2015). An overview of practical applications of protein disorder prediction and drive for faster, more accurate predictions. *Int. J. Mol. Sci*. 16: 15384–15404.

- Dong, Q., Wang, X., Lin, L., and Xu, Z. (2006). Domain boundary prediction based on profile domain linker propensity index. *Comput. Biol. Chem.* 30: 127–133.
- Dong, C., Wei, P., Jian, X. et al. (2015). Comparison and integration of deleteriousness prediction methods for nonsynonymous SNVs in whole exome sequencing studies. *Hum. Mol. Genet.* 24: 2125–2137.
- Eddy, S.R. (2011). Accelerated profile HMM searches. *PLoS Comput. Biol.* 7: e1002195.
- Elbarbary, R.A., Lucas, B.A., and Maquat, L.E. (2016). Retrotransposons as regulators of gene expression. *Science.* 351: aac7247.
- Esmailbeiki, R., Krawczyk, K., Knapp, B. et al. (2016). Progress and challenges in predicting protein interfaces. *Briefings Bioinf.* 17: 117–131.
- Eyrich, V., Martí-Renom, M.A., Przybylski, D. et al. (2001). EVA: continuous automatic evaluation of protein structure prediction servers. *Bioinformatics.* 17: 1242–1243.
- Ezkurdia, L., Grana, O., Izarzugaza, J.M.G., and Tress, M.L. (2009). Assessment of domain boundary predictions and the prediction of intramolecular contacts in CASP8. *Proteins Struct. Funct. Bioinf.* 77: 196–209.
- Fagerberg, L., Jonasson, K., and Heijne, G.V. (2010). Prediction of the human membrane proteome. *Proteomics.* 10: 1141–1149.
- Fariselli, P., Savojardo, C., Martelli, P.L., and Casadio, R. (2009). Grammatical-restrained hidden conditional random fields for bioinformatics applications. *Algorithms Mol. Biol.* 4: 13.
- Fidelis, K., Rost, B., and Zemla, A. (1999). A modified definition of Sov, a segment-based measure for protein secondary structure prediction assessment. *Proteins.* 223: 220–223.
- Finn, R.D., Bateman, A., Clements, J. et al. (2014a). Pfam: the protein families database. *Nucleic Acids Res.* 42: 222–230.
- Finn, R.D., Miller, B.L., Clements, J., and Bateman, A. (2014b). IPfam: a database of protein family and domain interactions found in the protein data Bank. *Nucleic Acids Res.* 42: 364–373.
- Finn, R.D., Attwood, T.K., Babbitt, P.C. et al. (2016). InterPro in 2017-beyond protein family and domain annotations. *Nucleic Acids Res.* 45: gkw1107.
- Fischer, T.B., Arunachalam, K.V., Bailey, D. et al. (2003). The binding interference database (BID): a compilation of amino acid hot spots in protein interfaces. *Bioinformatics.* 19: 1453–1454.
- Foster, L.J., de Hoog, C.L., Zhang, Y. et al. (2006). A mammalian organelle map by protein correlation profiling. *Cell.* 125 (1): 187–199.
- Frishman, D. and Argos, P. (1995). Knowledge-based protein secondary structure assignment. *Proteins Struct. Funct. Genet.* 23 (4): 566–579.
- Fukuchi, S., Amemiya, T., Sakamoto, S. et al. (2014). IDEAL in 2014 illustrates interaction networks composed of intrinsically disordered proteins and their binding partners. *Nucleic Acids Res.* 42: 320–325.
- Galperin, M.Y., Makarova, K.S., Wolf, Y.I., and Koonin, E.V. (2015). Expanded microbial genome coverage and improved protein family annotation in the COG database. *Nucleic Acids Res.* 43: D261–D269.
- Gardy, J.L. and Brinkman, F.S. (2006). Methods for predicting bacterial protein subcellular localization. *Nat. Rev. Microbiol.* 4 (10): 741–751.
- Garnier, J., Osguthorpe, D.J., and Robson, B. (1978). Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *J. Mol. Biol.* 120: 97–120.
- Garnier, J., Gibrat, J.-F., and Robson, B. (1996). GOR method for predicting protein secondary structure from amino acid sequence. *Methods Enzymol.* 266: 540–553.
- Garrow, A.G., Agnew, A., and Westhead, D.R. (2005). TMB-Hunt: a web server to screen sequence sets for transmembrane beta-barrel proteins. *Nucleic Acids Res.* 33 (Suppl 2): 188–192.
- Gaudet, P., Michel, P.A., Zahn-Zabal, M. et al. (2017). The neXtProt knowledgebase on human proteins: 2017 update. *Nucleic Acids Res.* 45 (D1): D177–D182.
- Gene Ontology Consortium (2000). Gene ontology: tool for the unification of biology. *Nat. Genet.* 25: 25–29.

- Gene Ontology Consortium (2015). Gene Ontology Consortium: going forward. *Nucleic Acids Res.* 43: D1049–D1056.
- Goldberg, T., Hamp, T., and Rost, B. (2012). LocTree2 predicts localization for all domains of life. *Bioinformatics.* 28: i458–i465.
- Goldberg, T., Hecht, M., Hamp, T. et al. (2014). LocTree3 prediction of localization. *Nucleic Acids Res.* 42 (Web Server issue): 1–6.
- Goodwin, S., McPherson, J.D., and McCombie, W.R. (2016). Coming of age: ten years of next-generation sequencing technologies. *Nat. Rev. Genet.* 17: 333–351.
- Gouw, M., Michael, S., Samano-Sanchez, H. et al. (2018). The eukaryotic linear motif resource – 2018 update. *Nucleic Acids Res.* 46 (D1): D428–D434.
- Graessel, A., Hauck, S.M., von Toerne, C. et al. (2015). A combined omics approach to generate the surface atlas of human naive CD4+ T cells during early T-cell receptor activation. *Mol. Cell. Proteomics.* 14 (8): 2085–2102.
- Greene, L.H., Lewis, T.E., Addou, S. et al. (2007). The CATH domain structure database: new protocols and classification levels give a more comprehensive resource for exploring evolution. *Nucleic Acids Res.* 35: 291–297.
- Grimm, D.G., Azencott, C.A., Aicheler, F. et al. (2015). The evaluation of tools used to predict the impact of missense variants is hindered by two types of circularity. *Hum. Mutat.* 36: 513–523.
- Habchi, J., Tompa, P., Longhi, S., and Uversky, V.N. (2014). Introducing protein intrinsic disorder. *Chem. Rev.* 114: 6561–6588.
- Haft, D.H., Selengut, J.D., Richter, R.A. et al. (2013). TIGRFAMs and genome properties in 2013. *Nucleic Acids Res.* 41: 387–395.
- Hamp, T. and Rost, B. (2012). Alternative protein-protein interfaces are frequent exceptions. *PLoS Comput. Biol.* 8 (8): e1002623.
- Hamp, T., Kassner, R., Seemayer, S. et al. (2013). Homology-based inference sets the bar high for protein function prediction. *BMC Bioinf.* 14 (Suppl 3): S7.
- Hayat, S., Peters, C., Shu, N. et al. (2016). Inclusion of dyad-repeat pattern improves topology prediction of transmembrane β -barrel proteins. *Bioinformatics.* 32: 1571–1573.
- Hecht, M., Bromberg, Y., and Rost, B. (2013). News from the protein mutability landscape. *J. Mol. Biol.* 425 (21): 3937–3948.
- Hecht, M., Bromberg, Y., and Rost, B. (2015). Better prediction of functional effects for sequence variants. *BMC Genomics.* 16 (Suppl 8): S1.
- Heffernan, R., Paliwal, K., Lyons, J. et al. (2015). Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning. *Sci. Rep.* 5: 11476.
- Heffernan, R., Yang, Y., Paliwal, K., and Zhou, Y. (2017). Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility. *Bioinformatics.* 33 (18): 2842–2849.
- von Heijne, G. (2006). Membrane-protein topology. *Nat. Rev. Mol. Cell Biol.* 7: 909–918.
- Heinig, M. and Frishman, D. (2004). STRIDE: a web server for secondary structure assignment from known atomic coordinates of proteins. *Nucleic Acids Res.* 32 (Web Server issue): 500–502.
- Hirose, S., Shimizu, K., Kanai, S. et al. (2007). Structural bioinformatics POODLE-L: a two-level SVM prediction system for reliably predicting long disordered regions. *Struct. Bioinf.* 23: 2046–2053.
- Hirose, S., Shimizu, K., and Noguchi, T. (2010). POODLE-I: disordered region prediction by integrating POODLE series and structural information predictors based on a work flow approach. *In Silico Biol.* 10: 185–191.
- Hönigschmid, P. (2012). *Improvement of DNA- and RNA-protein binding prediction.* Diploma thesis. TUM – Technical University of Munich.
- Hopf, T.A., Colwell, L.J., Sheridan, R. et al. (2012). Three-dimensional structures of membrane proteins from genomic sequencing. *Cell.* 149: 1607–1621.

- Hopf, T.A., Schärfe, C.P.I., Rodrigues, J.P.G.L.M. et al. (2014). Sequence co-evolution gives 3D contacts and structures of protein complexes. *eLife*. 3: e03430.
- Horton, P., Park, K.J., Obayashi, T. et al. (2007). WoLF PSORT: protein localization predictor. *Nucleic Acids Res.* 35 (Web Server issue): W585–W587.
- Hu, Y., Lehrach, H., and Janitz, M. (2009). Comparative analysis of an experimental subcellular protein localization assay and in silico prediction methods. *J. Mol. Histol.* 40 (5–6): 343–352.
- Hubbard, S.J. and Thornton, J.M. (1993). *NACCESS. Department of Biochemistry and Molecular Biology*. University College London.
- Huerta-Cepas, J., Szklarczyk, D., Forslund, K. et al. (2016). EGGNOG 4.5: a hierarchical orthology framework with improved functional annotations for eukaryotic, prokaryotic and viral sequences. *Nucleic Acids Res.* 44: D286–D293.
- Huh, W.K., Falvo, J.V., Gerke, L.C. et al. (2003). Global analysis of protein localization in budding yeast. *Nature*. 425 (6959): 686–691.
- Hwang, S., Guo, Z., and Kuznetsov, I.B. (2007). DP-bind: a web server for sequence-based prediction of DNA-binding residues in DNA-binding proteins. *Bioinformatics*. 23: 634–636.
- Ishida, T. and Kinoshita, K. (2007). PrDOS: prediction of disordered protein regions from amino acid sequence. *Nucleic Acids Res.* 35: 460–464.
- Ishida, T. and Kinoshita, K. (2008). Prediction of disordered regions in proteins based on the meta approach. *Bioinformatics*. 24 (11): 1344–1348.
- Jacoby, E., Bouhelal, R., Gerspacher, M., and Seuwen, K. (2006). The 7 TM G-protein-coupled receptor target family. *ChemMedChem*. 1: 761–782.
- Jensen, L.J. and Bateman, A. (2011). The rise and fall of supervised machine learning techniques. *Bioinformatics*. 27: 3331–3332.
- Jia, Y. and Liu, X.-Y. (2006). From surface self-assembly to crystallization: prediction of protein crystallization conditions. *J. Phys. Chem. B*. 110: 6949–6955.
- Jiang, Y., Oron, T.R., Clark, W.T. et al. (2016). An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biol* 17 (1): 184.
- Jones, D.T. (1999). Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.* 292: 195–202.
- Jones, D.T. and Cozzetto, D. (2015). DISOPRED3: precise disordered region predictions with annotated protein-binding activity. *Bioinformatics*. 31: 857–863.
- Jones, P., Binns, D., Chang, H.Y. et al. (2014). InterProScan 5: genome-scale protein function classification. *Bioinformatics*. 30: 1236–1240.
- Joo, K., Lee, S.J., and Lee, J. (2012). SANN: solvent accessibility prediction of proteins by nearest neighbor method. *Proteins* 80 (7): 1791–1797.
- Kabsch, W. and Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*. 22: 2577–2637.
- Kajan, L., Yachdav, G., Vicedo, E. et al. (2013). Cloud prediction of protein structure and function with PredictProtein for Debian. *Biomed. Res. Int.* 2013: 398968.
- Käll, L., Krogh, A., and Sonnhammer, E.L.L. (2004). A combined transmembrane topology and signal peptide prediction method. *J. Mol. Biol.* 338: 1027–1036.
- Käll, L., Krogh, A., and Sonnhammer, E.L.L. (2005). An HMM posterior decoder for sequence feature prediction that includes homology information. *Bioinformatics*. 21: i251.
- Keskin, O., Tuncbag, N., and Gursoy, A. (2016). Predicting protein-protein interactions from the molecular to the proteome level. *Chem. Rev.* 116: 4884–4909.
- Kessel, A. and Ben-Tal, N. (2011). *Introduction to Proteins*, 438–440. London, UK: CRC Press.
- Kihara, D. (2005). The effect of long-range interactions on the secondary structure formation of proteins. *Protein Sci.* 14: 1955–1963.
- Kinch, L.N., Li, W., Monastyrskyy, B. et al. (2016). Evaluation of free modeling targets in CASP11 and ROLL. *Proteins* 84 (Suppl 1): 51–66.
- Kircher, M., Witten, D.M., Jain, P. et al. (2014). A general framework for estimating the relative pathogenicity of human genetic variants. *Nat. Genet.* 46: 310–315.

- Klimke, W., Agarwala, R., Badretin, A. et al. (2009). The National Center for Biotechnology Information's protein clusters database. *Nucleic Acids Res.* 37: 216–223.
- Kloppmann, E., Punta, M., and Rost, B. (2012). Structural genomics plucks high-hanging membrane proteins. *Curr. Opin. Struct. Biol.* 22: 326–332.
- Köhler, S., Vasilevsky, N.A., Engelstad, M. et al. (2016). The human phenotype ontology in 2017. *Nucleic Acids Res.* 45: gkw1039.
- Krissinel, E. and Henrick, K. (2007). Inference of macromolecular assemblies from crystalline state. *J. Mol. Biol.* 372: 774–797.
- Krogh, A., Larsson, B., von Heijne, G., and Sonnhammer, E.L. (2001). Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. *J. Mol. Biol.* 305: 567–580.
- Kumar, M., Gromiha, M.M., and Raghava, G.P.S. (2008). Prediction of RNA binding sites in a protein using SVM and PSSM profile. *Proteins.* 71: 189–194.
- Kumar, P., Henikoff, S., and Ng, P.C. (2009). Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm. *Nat. Protoc.* 4: 1073–1081.
- Kyte, J. and Doolittle, R.F. (1982). A simple method for displaying the hydropathic character of a protein. *J. Mol. Biol.* 157: 105–132.
- Lam, S.D., Dawson, N.L., Das, S. et al. (2016). Gene3D: expanding the utility of domain assignments. *Nucleic Acids Res.* 44: D404–D409.
- de Las Rivas, J. and Fontanillo, C. (2010). Protein-protein interactions essentials: key concepts to building and analyzing interactome networks. *PLoS Comput. Biol.* 6: 1–8.
- Lee, B. and Richards, F.M. (1971). The interpretation of protein structures: estimation of static accessibility. *J. Mol. Biol.* 55 (3): 379–400.
- van der Lee, R., Buljan, M., Lang, B. et al. (2014). Classification of intrinsically disordered regions and proteins. *Chem. Rev.* 114: 6589–6631.
- Letunic, I., Doerks, T., and Bork, P. (2015). SMART: recent updates, new developments and status in 2015. *Nucleic Acids Res.* 43: D257–D260.
- Liu, J. and Rost, B. (2001). Comparing function and structure between entire proteomes. *Protein Sci.* 10: 1970–1979.
- Liu, J. and Rost, B. (2003). Domains, motifs and clusters in the protein universe. *Curr. Opin. Chem. Biol.* 7: 5–11.
- Liu, J. and Rost, B. (2004). CHOP proteins into structural domain-like fragments. *Proteins Struct. Funct. Genet.* 55: 678–688.
- Lobanov, M.Y. and Galzitskaya, O.V. (2015). How common is disorder? Occurrence of disordered residues in four domains of life. *Int. J. Mol. Sci.* 16: 19490–19507.
- Lobley, A. (2010). Human Protein Function Prediction: application of machine learning for integration of heterogeneous data sources. PhD thesis. University College London, LoNdon, UK.
- Magnan, C.N. and Baldi, P. (2014). SSpro/ACCpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity. *Bioinformatics.* 30: 2592–2597.
- Mahlich, Y., Reeb, J., Schelling, M. et al. (2017). Common sequence variants affect molecular function more than rare variants. *Sci. Rep.* 7: 1608.
- Marks, D.S., Colwell, L.J., Sheridan, R. et al. (2011). Protein 3D structure computed from evolutionary sequence variation. *PLoS One* 6: e28766.
- Marks, D.S., Hopf, T.A., Chris, S., and Sander, C. (2012). Protein structure prediction from sequence variation. *Nat. Biotechnol.* 30: 1072–1080.
- Martinez, D.A. and Nelson, M.A. (2010). The next generation becomes the now generation. *PLoS Genet.* 6: e1000906.
- Mi, H., Poudel, S., Muruganujan, A. et al. (2016). PANTHER version 10: expanded protein families and functions, and analysis tools. *Nucleic Acids Res.* 44: D336–D342.
- Miosge, L.A., Field, M.A., Sontani, Y. et al. (2015). Comparison of predicted and actual consequences of missense mutations. *Proc. Natl. Acad. Sci. USA.* 112: E5189–E5198.

- Mirabello, C. and Pollastri, G. (2013). Porter, PaleAle 4.0: high-accuracy prediction of protein secondary structure and relative solvent accessibility. *Bioinformatics*. 29 (16): 2056–2058.
- Monastyrskyy, B., Kryshchak, A., Moulton, J. et al. (2014). Assessment of protein disorder region predictions in CASP10. *Proteins Struct. Funct. Bioinf.* 82: 127–137.
- Montomerie, S., Sundararaj, S., Gallin, W.J., and Wishart, D.S. (2006). Improving the accuracy of protein secondary structure prediction using structural alignment. *BMC Bioinf.* 7: 301–301.
- Montomerie, S., Cruz, J.A., Shrivastava, S. et al. (2008). PROTEUS2: a web server for comprehensive protein structure prediction and structure-based annotation. *Nucleic Acids Res.* 36 (Web Server issue): 202–209.
- Mooney, C., Cessieux, A., Shields, D.C., and Pollastri, G. (2013). SCL-Epred: a generalised de novo eukaryotic protein subcellular localisation predictor. *Amino Acids*. 45 (2): 291–299.
- Morrow, J.K. and Zhang, S. (2012). Computational prediction of protein hot spot residues. *Curr. Pharm. Des.* 18: 1255–1265.
- Mosca, R., Céol, A., Stein, A. et al. (2014). 3did: a catalog of domain-based interactions of known three-dimensional structure. *Nucleic Acids Res.* 42: 374–379.
- Moulton, J., Pedersen, J.T., Judson, R., and Fidelis, K. (1995). A large-scale experiment to assess protein structure prediction methods. *Proteins Struct. Funct. Genet.* 23: ii–iv.
- Murakami, Y. and Mizuguchi, K. (2010). Applying the Naive Bayes classifier with kernel density estimation to the prediction of protein-protein interaction sites. *Bioinformatics*. 26: 1841–1848.
- Nair, R. and Rost, B. (2002). Inferring sub-cellular localisation through automated lexical analysis. *Bioinformatics*. 18 (Suppl 1): S78–S86.
- Necci, M., Piovesan, D., Dosztányi, Z., and Tosatto, S.C.E. (2017). MobiDB-lite: fast and highly specific consensus prediction of intrinsic disorder in proteins. *Bioinformatics*. 33: btx015.
- Ng, P.C. and Henikoff, S. (2003). SIFT: predicting amino acid changes that affect protein function. *Nucleic Acids Res.* 31: 3812–3814.
- Nugent, T. and Jones, D.T. (2009). Transmembrane protein topology prediction using support vector machines. *BMC Bioinf.* 10: 159.
- Nugent, T. and Jones, D.T. (2012). Accurate de novo structure prediction of large transmembrane protein domains using fragment-assembly and correlated mutation analysis. *Proc. Natl. Acad. Sci. USA* 109: E1540–E1547.
- Oates, M.E., Romero, P., Ishida, T. et al. (2013). D2P2: database of disordered protein predictions. *Nucleic Acids Res.* 41: 508–516.
- Oates, M.E., Stahlhacke, J., Vavoulis, D.V. et al. (2015). The SUPERFAMILY 1.75 database in 2014: a doubling of data. *Nucleic Acids Res.* 43: D227–D233.
- O'Donovan, C., Martin, M.J., Gattiker, A. et al. (2002). High-quality protein knowledge resource: SWISS-PROT and TrEMBL. *Briefings Bioinf.* 3: 275–284.
- Ofran, Y. and Rost, B. (2003a). Analysing six types of protein-protein interfaces. *J. Mol. Biol.* 325: 377–387.
- Ofran, Y. and Rost, B. (2003b). Predicted protein-protein interaction sites from local sequence information. *FEBS Lett.* 544: 236–239.
- Ofran, Y. and Rost, B. (2007). ISIS: interaction sites identified from sequence. *Bioinformatics*. 23 (2): e13–e16.
- Overington, J., Al-Lazikani, B., and Hopkins, A.L. (2006). How many drug targets are there? *Nat. Rev. Drug Discov.* 5: 993–996.
- Pang, C.I., Lin, K., Wouters, M.A. et al. (2008). Identifying foldable regions in protein sequence from the hydrophobic signal. *Nucleic Acids Res.* 36: 578–588.
- Pawson, T. and Nash, P. (2003). Assembly of cell regulatory systems through protein interaction domains. *Science*. 300 (5618): 445–452.
- Pedruzzi, I., Rivoire, C., Auchincloss, A.H. et al. (2013). HAMAP in 2013, new developments in the protein family classification and annotation system. *Nucleic Acids Res.* 41: 584–589.
- Pedruzzi, I., Rivoire, C., Auchincloss, A.H. et al. (2015). HAMAP in 2015: updates to the protein family classification and annotation system. *Nucleic Acids Res.* 43: D1064–D1070.

- Piovesan, D., Tabaro, F., Mičetić, I. et al. (2016). DisProt 7.0: a major update of the database of disordered proteins. *Nucleic Acids Res.* 45: gkw1056.
- Pollastri, G., Przybylski, D., Rost, B., and Baldi, P. (2002). Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins Struct. Funct. Bioinf.* 47: 228–235.
- Punta, M. and Rost, B. (2008). Neural networks predict protein structure and function. *Methods Mol. Biol.* 458: 203–230.
- Radivojac, P., Clark, W.T., Oron, T.R. et al. (2013). A large-scale evaluation of computational protein function prediction. *Nat. Methods.* 10: 221–227.
- Ramilowski, J.A., Goldberg, T., Harshbarger, J. et al. (2015). A draft network of ligand-receptor-mediated multicellular signalling in human. *Nat. Commun.* 6: 7866.
- Rao, V.S., Srinivas, K., Sujini, G.N., and Kumar, G.N.S. (2014). Protein-protein interaction detection: methods and analysis. *Int. J. Proteomics.* 2014: 1–12.
- Reeb, J., Kloppmann, E., Bernhofer, M., and Rost, B. (2014). Evaluation of transmembrane helix predictions in 2014. *Proteins Struct. Funct. Bioinf.* 83: 473–484.
- Reeb, J., Hecht, M., Mahlich, Y. et al. (2016). Predicted molecular effects of sequence variants link to system level of disease. *PLoS Comput. Biol.* 12 (8): e1005047.
- Remmert, M., Biegert, A., Hauser, A., and Söding, J. (2012). HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nat. Methods.* 9 (2): 173–175.
- Res, I., Mihalek, I., and Lichtarge, O. (2005). An evolution based classifier for prediction of protein interfaces without using protein structures. *Bioinformatics.* 21 (10): 2496–2501.
- Rezáčová, P., Borek, D., Moy, S.F. et al. (2008). Crystal structure and putative function of small Toprim domain-containing protein from *Bacillus stearothermophilus*. *Proteins.* 70: 311–319.
- Rose, P.W., Prlić, A., Altunkaya, A. et al. (2017). The RCSB protein data bank: integrative view of protein, gene and 3D structural information. *Nucleic Acids Res.* 45 (D1): D271–D281.
- Rost, B. (1996). PHD: predicting one-dimensional protein structure by profile based neural networks. *Methods Enzymol.* 266: 525–539.
- Rost, B. (2001). Protein secondary structure prediction continues to rise. *J. Struct. Biol.* 134: 204–218.
- Rost, B. (2002). Enzyme function less conserved than anticipated. *J. Mol. Biol.* 318: 595–608.
- Rost, B. and Sander, C. (1993). Improved prediction of protein secondary structure by use of sequence profiles and neural networks. *Proc. Natl. Acad. Sci. USA* 90: 7558–7562.
- Rost, B. and Sander, C. (1994a). Combining evolutionary information and neural networks to predict protein secondary structure. *Proteins Struct. Funct. Bioinf.* 19: 55–72.
- Rost, B. and Sander, C. (1994b). Conservation and prediction of solvent accessibility in protein families. *Proteins Struct. Funct. Genet.* 20 (3): 216–226.
- Rost, B., Yachdav, G., and Liu, J. (2004). The PredictProtein server. *Nucleic Acids Res.* 32 (Suppl 2): W321–W326.
- Rychlewski, L. and Fischer, D. (2005). LiveBench-8: the large-scale, continuous assessment of automated protein structure prediction. *Protein Sci.* 14 (1): 240–245.
- Savojardo, C., Fariselli, P., and Casadio, R. (2013). BETAWARE: a machine-learning tool to detect and predict transmembrane beta-barrel proteins in prokaryotes. *Bioinformatics.* 29: 504–505.
- Schaarschmidt, J., Monastyrskyy, B., Kryshtafovych, A., and Bonvin, A.M.J.J. (2018). Assessment of contact predictions in CASP12: co-evolution and deep learning coming of age. *Proteins* 86 (Suppl 1): 51–66.
- Schlessinger, A., Schaefer, C., Vicedo, E. et al. (2011). Protein disorder – a breakthrough invention of evolution? *Curr. Opin. Struct. Biol.* 21: 412–418.
- Schrodinger LLC. (2015). The PyMOL Molecular Graphics System, Version 1.9.
- Shimizu, K., Hirose, S., and Noguchi, T. (2007). Structural bioinformatics POODLE-S: web application for predicting protein disorder by using physicochemical features and reduced amino acid set of a position-specific scoring matrix. *Struct. Bioinf.* 23: 2337–2338.

- Shoemaker, B.A., Zhang, D., Tyagi, M. et al. (2012). IBIS (inferred biomolecular interaction server) reports, predicts and integrates multiple types of conserved interactions for proteins. *Nucleic Acids Res.* 40: 834–840.
- Sigrist, C.J.A., De Castro, E., Cerutti, L. et al. (2013). New and continuing developments at PROSITE. *Nucleic Acids Res.* 41: 344–347.
- Šikić, M., Tomić, S., and Vlahoviček, K. (2009). Prediction of protein-protein interaction sites in sequences and 3D structures by random forests. *PLoS Comput. Biol.* 5 (1): e1000278.
- Sillitoe, I., Cuff, A.L., Dessailly, B.H. et al. (2013). New functional families (FunFams) in CATH to improve the mapping of conserved functional sites to 3D structures. *Nucleic Acids Res.* 41: 490–498.
- Sillitoe, I., Lewis, T.E., Cuff, A. et al. (2015). CATH: comprehensive structural and functional annotations for genome sequences. *Nucleic Acids Res.* 43: D376–D381.
- Söding, J. (2005). Protein homology detection by HMM-HMM comparison. *Bioinformatics.* 21: 951–960.
- Stevens, T.J. and Arkin, I.T. (2000). Do more complex organisms have a greater proportion of membrane proteins in their genomes? *Proteins* 39: 417–420.
- Suyama, M. and Ohara, O. (2003). DomCut: prediction of inter-domain linker regions in amino acid sequences. *Bioinformatics.* 19: 673–674.
- Szent-Györgyi, A.G. and Cohen, C. (1957). Role of proline in polypeptide chain configuration of proteins. *Science.* 126: 697.
- Thorn, K.S. and Bogan, A.A. (2001). ASEdb: a database of alanine mutations and their effects on the free energy of binding in protein interactions. *Bioinformatics.* 17: 284–285.
- Thusberg, J., Olatubosun, A., and Vihinen, M. (2011). Performance of mutation pathogenicity prediction methods on missense variants. *Hum. Mutat.* 32: 358–368.
- Tien, M.Z., Meyer, A.G., Sydykova, D.K. et al. (2013). Maximum allowed solvent accessibilities of residues in proteins. *PLoS One.* 8 (11): e80635.
- Tompa, P., Davey, N.E., Gibson, T.J., and Babu, M.M. (2014). A million peptide motifs for the molecular biologist. *Mol. Cell.* 55 (2): 161–169.
- Touw, W.G., Baakman, C., Black, J. et al. (2015). A series of PDB-related databanks for everyday needs. *Nucleic Acids Res.* 43 (D1): D364–D368.
- Tsirigos, K.D., Elofsson, A., and Bagos, P.G. (2016). PRED-TMBB2: improved topology prediction and detection of beta-barrel outer membrane proteins. *Bioinformatics.* 32 (17): i665–i671.
- Tuncbag, N., Kar, G., Keskin, O. et al. (2009). A survey of available tools and web servers for analysis of protein-protein interactions and interfaces. *Briefings Bioinf.* 10: 217–232.
- UniProt Consortium (2016). UniProt: the universal protein knowledgebase. *Nucleic Acids Res.* 45: 1–12.
- Vicedo, E., Schlessinger, A., and Rost, B. (2015). Environmental pressure may change the composition protein disorder in prokaryotes. *PLoS One.* 10: 1–21.
- Viklund, H., Granseth, E., and Elofsson, A. (2006). Structural classification and prediction of reentrant regions in alpha-helical transmembrane proteins: application to complete genomes. *J. Mol. Biol.* 361: 591–603.
- Von Heijne, G. (1992). Membrane protein structure prediction. Hydrophobicity analysis and the positive-inside rule. *J. Mol. Biol.* 225: 487–494.
- Von Heijne, G. and Gavel, Y. (1988). Topogenic signals in integral membrane proteins. *Eur. J. Biochem.* 174: 671–678.
- Walia, R.R., Xue, L.C., Wilkins, K. et al. (2014). RNABindRPlus: a predictor that combines machine learning and sequence homology-based methods to improve the reliability of predicted RNA-binding residues in proteins. *PLoS One* 9 (5): e97725.
- Wang, B., Chen, P., Huang, D.S. et al. (2006). Predicting protein interaction sites from residue spatial sequence profile and evolution rate. *FEBS Lett.* 580: 380–384.
- Wang, S., Li, W., Liu, S., and Xu, J. (2016a). RaptorX-property: a web server for protein structure property prediction. *Nucleic Acids Res.* 44 (W1): W430–W435.

- Wang, S., Peng, J., Ma, J., and Xu, J. (2016b). Protein secondary structure prediction using deep convolutional neural fields. *Sci. Rep.* 6: 18962.
- Wang, S., Sun, S., Li, Z. et al. (2017). Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLoS Comput. Biol.* 13 (1): e1005324.
- Wright, P.E. and Dyson, H.J. (2014). Intrinsically disordered proteins in cellular signalling and regulation. *Nat. Rev. Mol. Cell Biol.* 16: 18–29.
- Wu, C.H., Nikolskaya, A., Huang, H. et al. (2004). PIRSF: family classification system at the protein information resource. *Nucleic Acids Res.* 32: D112–D114.
- Xue, L.C., Dobbs, D., and Honavar, V. (2011). HomPPI: a class of sequence homology based protein-protein interface prediction methods. *BMC Bioinf.* 12: 244.
- Yachdav, G., Kloppmann, E., Kajan, L. et al. (2014). PredictProtein – an open resource for online prediction of protein structural and functional features. *Nucleic Acids Res.* 42: W337–W343.
- Yan, J., Friedrich, S., and Kurgan, L. (2016). A comprehensive comparative review of sequence based predictors of DNA- and RNA-binding residues. *Briefings Bioinf.* 17: 88–105.
- Yang, Y., Gao, J., Wang, J. et al. (2016a). Sixty-five years of the long march in protein secondary structure prediction: the final stretch. *Briefings Bioinf.* 19 (3): 482–494.
- Yang, J., Jin, Q.Y., Zhang, B., and Shen, H.B. (2016b). R2C: improving ab initio residue contact map prediction using dynamic fusion strategy and Gaussian noise filter. *Bioinformatics.* 32: 2435–2443.
- Zhang, H., Zhang, T., Chen, K. et al. (2011). Critical assessment of high-throughput standalone methods for secondary structure prediction. *Briefings Bioinf.* 12 (6): 672–688.
- Zhao, H., Yang, Y., and Zhou, Y. (2013). Prediction of RNA binding proteins comes of age from low resolution to high resolution. *Mol. Biosyst.* 9: 2417–2425.

8

Multiple Sequence Alignments

Fabian Sievers, Geoffrey J. Barton, and Desmond G. Higgins

Introduction

A multiple sequence alignment (MSA) is an arrangement of more than two amino acid or nucleotide sequences which are aligned so as to make the residues from the different sequences line up in vertical columns in some appropriate manner. These are used in a great variety of analyses and pipelines in proteome and genome analysis and are an essential initial step in most phylogenetic comparisons. They are widely used to help search for common features in sequences and can be used to help predict two- and three-dimensional structures of proteins and nucleic acids. An excellent review of MSA methods, uses, and abuses is provided by Chatzou et al. (2016).

Usually, one should only attempt to align sequences which are phylogenetically related and, therefore, homologous. In this case, the ideal alignment will have homologous residues aligned in the columns. An example of a multiple protein sequence alignment is shown in Figure 8.1. Here, one column is highlighted. If this column is well aligned, one can infer that the residues in that column have been derived from the same residue in the common ancestor of these sequences. That residue could have been a valine (V) or an isoleucine (I) or some other residue, but the key thing is that all of the amino acids in that column derive from that one position in the common ancestor. This is the phylogenetic perspective that underlies the construction of these alignments. In principle, one could also attempt to align the sequences so as to maximize the structural, functional, or physicochemical similarity of the residues in each column. In simple cases, if the sequences are homologous, a good phylogenetic alignment will also maximize structural similarity. If the sequences are not homologous or so highly divergent that similarity is not clear, then a functional alignment may be very difficult to achieve. One common example of this kind of difficulty involves promoter sequences that share short functional motifs, such as binding sites for regulatory proteins. Most MSA packages struggle to correctly align such motifs and these are best searched for using special motif-finding packages or by comparison with sets of known motifs. A second example is where protein sequences share a common fold but no sequence similarity, perhaps because of convergent evolution of their three-dimensional structures or because of extreme divergence of the sequences. Again, such alignments are best carried out using special sequence–structure matching packages. In this chapter, we focus specifically on cases where we wish to align sequences that are clearly homologous and phylogenetically related.

When constructing an MSA, one must also take into account insertions and deletions that have taken place in the time during which the sequences under consideration have diverged from one another, after gene duplication or divergence of the host species. This means that MSA packages have to be able to find an arrangement of null characters or “gaps” that will somehow maximize the alignment of homologous residues in a fashion similar to that done for pairwise sequence alignments, as discussed in Chapter 3. These gaps are frequently represented by hyphens, as shown in Figure 8.1. Given a scoring scheme for residue matches

```

HBB_HUMAN      -----VHLTPEEKSAVTALWGKVN--VDEVGGEALGRLLVVYPWTQRFEFESFGDLST
HBB_HORSE      -----VQLSGEEKAAVLALWDKVN--EEEVGGGEALGRLLVVYPWTQRFDFSFGDLSN
HBA_HUMAN      -----VLSPADKTNVKAAWGKVGGAHAGEYGAEALERMFSLFPPTTKTYFPHF-DLS-
HBA_HORSE      -----VLSAADKTNVKAAWSKVGGHAGEYGAEALERMFLGFPPTTKTYFPHF-DLS-
GLB5_PETMA     PIVDTGSVAPLSAAEKTIRSAWAPVYSTYETSGVDILVKFFTSTPAAQEFFPKFKGLTT
MYG_PHYCA      -----VLSEGEWQLVLHVWAKVEADVAGHGQDILIRLFKSHPETLEKFRFKHLKT
LGB2_LUPLU     -----GALTESQAALVKSSWEFNANIPKHTHRFFILVLEIAPAAKDLFSFLKGTSE
                *:  :  :  *  .  :  .:  *  :  *  :  .

HBB_HUMAN      PDAVMGNPKVKAHGKLVVIGAFSDGLAHLDN-----LKGTFAATLSELHCDKLVDPENFRL
HBB_HORSE      PGAVMGNPKVKAHGKLVVLSFSGEGVHHLDN-----LKGTFAALSELHCDKLVDPENFRL
HBA_HUMAN      ----HGSAQVKGHGKLVADALTNVAHVHDD-----MPNALSALSDLHAHKLRVDPVNFKL
HBA_HORSE      ----HGSAQVKAHGKLVSDALTLAVGHLLD-----LPGALSNSLDLHAHKLRVDPVNFKL
GLB5_PETMA     ADQLKKSADVRWHAERITNAVNDAVASMDDT--EKMSMKLRDLGSKHAKSFQVDPQYFKV
MYG_PHYCA      EAEMKASEDLKKHGVTITALGAILKKGKGH-----HEAELKPLAQSHATKHKIPIKYLEF
LGB2_LUPLU     VP--QNNPELQAHAGVIVFKLVYEAAIQLVQVTGVVVTDATLKNLGSVHVSKG-VADAHFPV
                . . : : * . : . : . : * . * . : : : .

HBB_HUMAN      LGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH-----
HBB_HORSE      LGNVLVVVLARHFGKDFTPPELQASYQKVVAGVANALAHKYH-----
HBA_HUMAN      LSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTISKYR-----
HBA_HORSE      LSHCLLVTLAVHLPNDFTPAVHASLDKFLSSVSTVLTISKYR-----
GLB5_PETMA     LAAVIADTVAAG-----DAGFEKLSMCIILLRSAY-----
MYG_PHYCA      ISEAI IHVLSRHPGDFGADAQGAMNKALELFRKDI AAKYKELGYQG
LGB2_LUPLU     VKEAIIKTIKEVVGAKWSEELNSAWTIAYDELAIVIKKEMNDAA---
                :  :  .:  . . .  . :

```

Figure 8.1 An example multiple sequence alignment of seven globin protein sequences. One position is highlighted.

(e.g. BLOSUM62; Henikoff and Henikoff 1992) and scores for gaps, one can attempt to find an MSA that produces the best overall score (and, thereby, the best overall alignment). In principle, this can be done using extensions of dynamic programming sequence alignment methods (Needleman and Wunsch 1970) to many sequences. This would then guarantee the best-scoring MSA. In practice, such extensions require time and memory that involve an exponential function of the number of sequences (written $O(L^N)$, for N sequences of length L) and are limited to tiny numbers of sequences. Therefore, all of the methods that are widely used rely on *heuristics* to make the MSAs. The use of heuristics makes very large alignments possible but comes at the expense of a lack of guarantees about alignment scores or quality.

The most widely used MSA heuristic was called “progressive alignment” by Feng and Doolittle (1987); this method also belongs to a family of methods that were described by different groups in the 1980s (see, for example, Hogeweg and Hesper 1984). The earliest automatic MSA method that we are aware of was described by David Sankoff in 1973 (Sankoff et al. 1973) for aligning 5S rRNA sequences and is essentially a form of progressive alignment. All of these methods work by starting with alignments of pairs of sequences and merging these with new sequences or alignments to build up the MSA progressively. The order in which these alignments are performed is usually done according to some form of clustering of the sequences, generated by an all-against-all comparison, referred to as a “guide tree” in Higgins et al. (1992). A generic outline of this process is illustrated in Figure 8.2.

Measuring Multiple Alignment Quality

There are literally hundreds of different MSA packages and each uses different combinations of parameter settings and heuristic algorithms to make the alignments. How can we tell which package works best or is best-suited to which kinds of data? One standard approach is to compare alignments produced by different packages with a set of established “gold standard” reference alignments. Such sets are used as benchmarks and have been invaluable for developers of MSA packages in order to test and compare MSAs. For proteins, the most widely

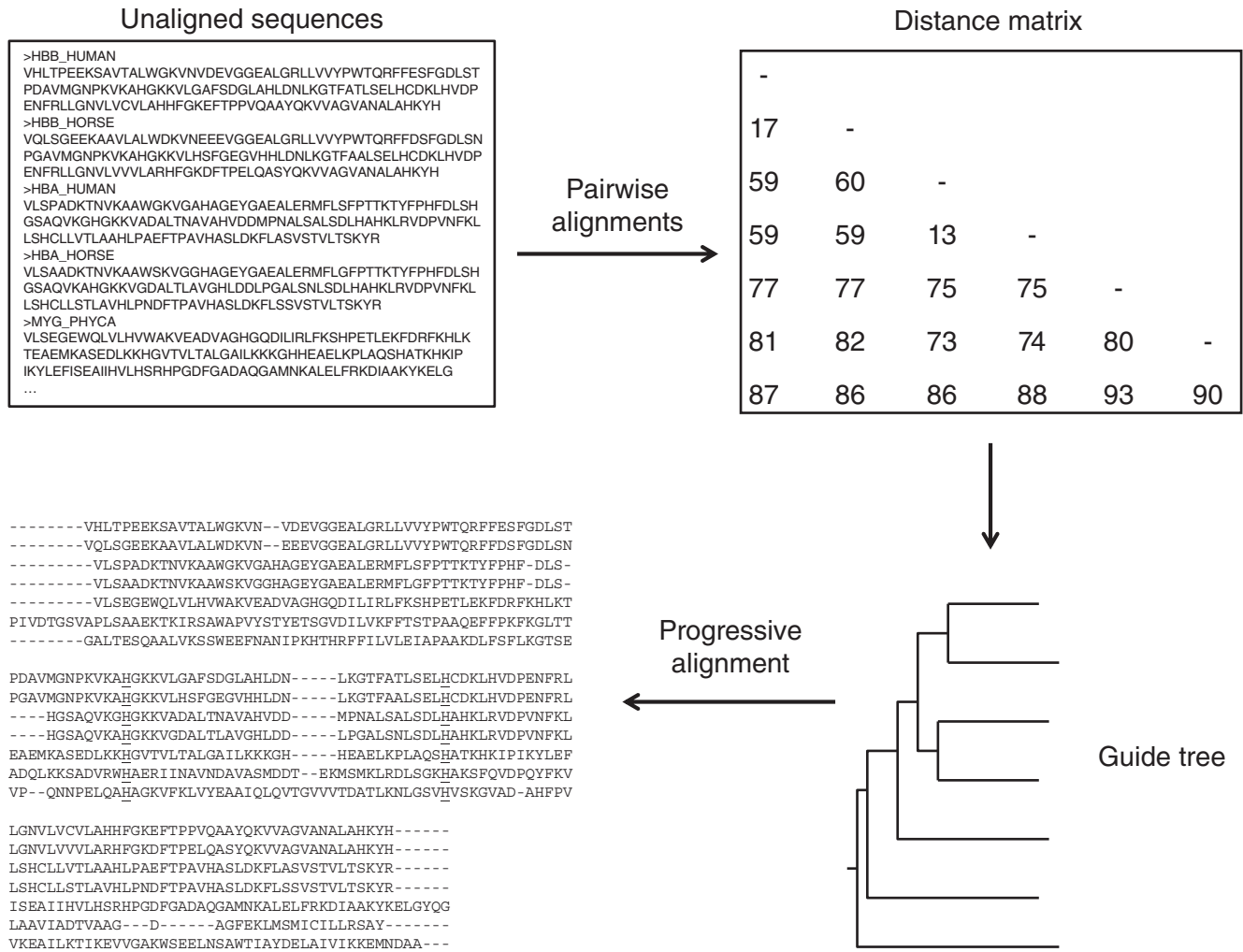


Figure 8.2 An outline of the simple progressive multiple alignment process. There are variations for all of these steps and some of them are iterated in well-known packages such as MAFFT and MUSCLE.

used MSA benchmarks have tended to rely on comparisons of protein sequences with known structures. This is due to the observation that protein sequences with very similar structures can actually have very highly divergent sequences. Therefore, this approach is very much based on a *structural* perspective. In turn, *phylogenetic* benchmarks have tended to use simulated alignments and/or sets of sequences with known phylogeny and do not necessarily give the same results as one would obtain based on structure (Iantorno et al. 2014). The use of structures in a benchmark entails aligning the structures automatically or manually, then using the corresponding sequence alignment to test various MSA packages. Early structural aligners included SSAP (Taylor and Orengo 1989) and STAMP (Russell and Barton 1992); a more recent program is MUSTANG (Konagurthu et al. 2006). While this process leads to a structural superposition of extant sections of the sequences to be aligned, it may not always be easy to align individual residues. Therefore, creating a reliable reference alignment may require some manual intervention, something that is not always straightforward (Edgar 2010; Iantorno et al. 2014).

The earliest large-scale MSA benchmark is BALiBASE. The original version (Thompson et al. 1999) contained over 140 reference alignments divided into five hierarchical reference sets, in an attempt to cover many different alignment scenarios. These include equidistant sequences of similar length (the BB11/12 reference set), families containing orphan sequences (BB2), equidistant divergent families (BB3), N/C-terminal extensions (BB4), and alignments with

insertions (BB5). For categories BB11, BB12, BB2, and BB3, different sequence lengths from less than 100 to more than 400 residues are covered. For category BB11/12, alignments with low to high sequence identity are used. While the current version of BALiBASE is 4.0, we will use BALiBASE 3.0 (Thompson et al. 2005) for the purposes of this discussion. Version 3 comprises the same five categories as version 1; however, the number of reference alignments has been increased to 218. The number of sequences of the reference alignments ranges from 4 to 142, with a median of 21. The BALiBASE benchmark contains a scoring program that assesses how well a generated (test) protein MSA resembles the reference alignment. Similarity between test and reference alignments in BALiBASE is expressed by two numbers: the sum-of-pairs (SP) score and the total column (TC) score. The scoring program measures SP and TC scores only for regions that are reliably aligned in the reference; these regions are the so-called “core columns.” OXBench (Raghava et al. 2003) and SABmark (Van Walle et al. 2005) are based on similar principles as BALiBASE. SABmark comprises 1268 alignments, ranging from three to 50 (median eight) sequences. OXBench comprises 672 families of between two and 122 (median three) sequences. In this chapter, we give SP and TC scores for various MSA packages, as measured using the BALiBASE benchmark.

The SP score measures the proportion of correctly aligned residue pairs while the TC score measures the proportion of reference alignment columns, which are perfectly retrieved in the generated MSA. Both scores can vary between 0 (that is, no residue pair or column retrieved) and 1 (that is, the generated MSA and the reference alignment are identical). For a pairwise alignment, the SP score and TC score are the same. For an MSA aligning three or more sequences, the TC score can never exceed the SP score. The SP and TC scores give a measure of the sensitivity of the aligner, measuring the fraction of correctly aligned residues and columns (the number of true positives). They do not, however, penalize for incorrectly aligned residues, which would be a measure of the specificity of the aligner (the number of true negatives). The specificity and sensitivity (see Box 5.4) of alignments in a benchmark test can be quantified by the Cline shift score (Cline et al. 2002) and the QModeler score (Sauder et al. 2000), which take incorrectly aligned residues into account.

The maximum number of sequences in benchmarks such as BALiBASE 3.0, SABmark, or OXBench is of the order of 100. None of these benchmarks can explore the performance of MSA software if thousands or even millions of sequences have to be aligned. One way to increase the number of sequences that can be aligned is to mix a set of sequences for which a reliable alignment is known with sequences for which no reliable alignment is known. This was done with OXBench to give the “extended dataset,” which had datasets of over 1000 sequences for some families. PREFAB (Edgar 2004) was designed from the outset with this principle in mind. PREFAB comprises 1682 reference alignments of two sequences, to which between 0 and 48 (median 48, mean 45.2) non-reference sequences are added. The software performs the alignment of the full set of (up to 50) sequences. However, the quality of the alignment can only be evaluated based on the alignment of the two reference sequences. A general purpose scoring program called qscore is available from the same web site that distributes PREFAB and MUSCLE.

A benchmark that extends the number of sequences into the tens of thousands is HomFam (Blackshields et al. 2010; Sievers et al. 2013). It is based on similar principles to PREFAB in that it mixes a small number of sequences for which a reliable alignment is known with a large number of homologous sequences for which no reliable alignment is known. The reference alignments come from the Homstrad structure alignment database (Mizuguchi et al. 1998) and the bulk of the sequences come from Pfam (Finn et al. 2014). The reference alignments comprise between five and 41 sequences, while the number of Pfam sequences varies between approximately 100 and 100 000. The 2013 HomFam dataset contains 95 families.

Recently, a new class of benchmarks has been devised that can test an aligner with arbitrarily large numbers of sequences, relies on a small number of references, and assesses the alignment of all sequences in the alignment (including non-reference sequences). The first such benchmark is ContTest (Fox et al. 2016). In ContTest, the MSA is used to detect the co-evolution of

alignment columns and produce a contact map prediction (Marks et al. 2011). This contact map prediction is then compared with the observed contact map of an embedded reference sequence. The accuracy with which the predicted and the observed contact maps agree serves as a proxy for the alignment quality. Co-evolution can only be detected if the information content of the alignment is large enough; that is, there should be at least as many sequences in the alignment as there are residues in the reference sequences. In practice, the number of sequences should be five times as large, so, for a typical protein domain, ContTest will not work well for fewer than 1000 sequences.

Another such benchmark is QuanTest (Le et al. 2017). Here, the MSA is used to predict secondary structure (Drozdetskiy et al. 2015), and then this predicted secondary structure is compared with the true secondary structure of one or more of the embedded reference sequences. In general, secondary structure prediction accuracy increases with the number of aligned sequences, but useful predictions can already be made for 200 sequences. Therefore, QuanTest is more applicable to smaller alignments than ContTest.

Making an Alignment: Practical Issues

Most automatic alignment programs such as the ones described in the next section will give good quality alignments for sequences that are similar. However, building good multiple alignments for sequences that are highly divergent is an expert task even with the best available alignment tools. In this section we give an overview of some of the steps to go through in order to make alignments that are good for structure/function prediction. This is not a universal recipe, as each set of sequences presents its own problems and only experience can guide the creation of high-quality alignments.

The key steps in building a multiple alignment are as follows:

- Find the sequences to align by database searching or other means.
- Locate the region(s) of each sequence to include in the alignment. Do not try to multiply align sequences that are very different in length. Most multiple alignment programs are designed to align sequences that are similar over their entire length, so first edit the sequences down to those regions that the sequence database search suggests are similar. Some database search tools can be of assistance in identifying such regions (e.g. PSI-BLAST; Altschul et al. 1997).
- Run the multiple alignment program.
- Inspect the alignment for problems. Take particular care over regions that appear to be speckled with gaps. Use an alignment visualization tool (e.g. Jalview or SeaView; see Viewing a Multiple Alignment) to identify positions in the alignment that conserve physicochemical properties across the complete alignment. If there are no such regions, then look at subsets of the sequences.
- Remove sequences that appear to seriously disrupt the alignment and then realign the subset that is left.
- After identifying key residues in the set of sequences that are straightforward to align, attempt to add the remaining sequences to the alignment so as to preserve the key features of the family.

With the exception of the first step (database search), all of the above steps can be managed within the Jalview program (see Viewing a Multiple Alignment), software that combines powerful alignment editing and subsetting functions with integrated access to eight multiple alignment algorithms. Alternatively, many of the programs described below can be run from web sites where the user pastes a set of sequences into a window or uploads a file with sequences in a standard file format. This works well for occasional use, and the use of many of these web sites is relatively self-explanatory. In particular, we recommend the tools server at the European Bioinformatics Institute (EBI), which allows for online usage of the most widely

used MSA packages. Some servers have limits on the number of sequences that can be aligned at one time or the user may need to make hundreds of alignments. In these cases, the user can run these alignment programs locally on a server or a desktop computer. Familiarity with the basic use of the Linux operating system then becomes important. All of the commonly used alignment packages can be run using so-called “command-line input,” where the user enters the name of the program (e.g. `clustalo`) at a prompt in a terminal window, followed by instructions for input and output. Basic usage for Linux command-line operation is given below for most of the commonly used multiple alignment packages.

Commonly Used Alignment Packages

Here, we describe how to make multiple alignments using a range of commonly used packages. Summary information for downloading source code or for online usage is given in Internet Resources.

Clustal Omega

Clustal Omega (Sievers et al. 2011) is the latest installment of the Clustal MSA suite for use with amino acid and nucleotide sequences. It is an almost complete rewrite of its predecessor, ClustalW2 (Larkin et al. 2007). The main improvements over ClustalW2 are that Clustal Omega can align much larger numbers of sequences than ClustalW2 in a shorter amount of time, producing alignments that are usually more accurate as measured by crystal structure-based benchmarks, and that it can incorporate prior knowledge about the general structure of the final alignment. Clustal Omega is a command-line-driven program that has been successfully compiled for Linux, Mac, and Windows. Unlike its predecessor, Clustal Omega does not have a graphical user interface (GUI), but this absence is mitigated by the existence of many very good alignment visualization programs such as SeaView (Gouy et al. 2010) and Jalview (Waterhouse et al. 2009), as well as by online web servers such as the European Molecular Biology Laboratory (EMBL)-EBI bioinformatic web and programmatic tools framework, the Max Planck Bioinformatics Toolkit, and the Galaxy server of the Pasteur Institute.

Clustal Omega is a progressive aligner. A “guide tree” is used to guide the multiple alignment; this guide tree is calculated from a matrix of pairwise distances among the sequences. For N sequences, this requires $N \times N$ sequence comparisons and the storage of an $N \times N$ distance matrix. In the past, this step was usually the bottleneck that prevented conventional aligners from aligning large numbers of sequences. Practical limits were of the order of 10 000 sequences or fewer. However, and by default, Clustal Omega does not calculate an all-against-all distance matrix, but uses the mBed algorithm instead (Blackshields et al. 2010). mBed calculates a distance matrix of all sequences against a small number of randomly chosen “seed” sequences. The computational requirements of the mBed algorithm, therefore, do not scale quadratically with N but rather as $N \times \log(N)$. Clustal Omega uses the mBed distance matrix to perform a k -means clustering of the sequences. By default, the cluster sizes have an upper limit of 100 sequences. Small guide trees are generated for the clusters, and an overarching guide tree is constructed for the clusters. The default upper cluster size was set to 100 when typical alignment sizes did not routinely exceed 10 000, such that there would be at most 100 clusters of size 100; for larger alignments, the cluster size can be adjusted by setting the `--cluster-size` flag. Despite the apparent reduction in information owing to a smaller distance matrix, alignments generated using an mBed guide tree are usually of equal quality to (if not higher quality than) an all-against-all-based distance matrix. The mBed mode can be turned off using the `--full` flag for a full distance matrix calculation.

In the main alignment step of the progressive alignment heuristic, individual sequences are aligned to form subalignments, and small subalignments are aligned to each other to form larger and larger subalignments. These pairwise alignments are carried out in Clustal

Omega using `hhalgn` (Söding 2005). This program converts individual sequences and small subalignments into hidden Markov models (HMMs), then aligns these HMMs in a pairwise fashion.

Clustal Omega's file input/output process uses Sean Eddy's squid library, allowing it to read and write several widely used sequence formats such as a2m/FASTA, Clustal, msf, PHYLIP, selex, Stockholm, and Vienna. The default output format is FASTA. A minimum Clustal Omega command line would be written as follows:

```
clustalo -i <infile> -o <outfile>
```

where `<infile>` is a placeholder for a file containing sequences to be aligned in one of the recognized file formats and `<outfile>` is a placeholder for the file where the aligned sequences will be stored in FASTA format.

Iteration Clustal Omega has the ability to iteratively refine an alignment. In the initial alignment phase, distances are based on k -mers of unaligned sequences. During the iterative refinement, distances will be based on a full alignment. The hope is that these full alignment distances are a better reflection of the sequences' similarity and will, therefore, produce a "better" guide tree that will, in turn, produce a better alignment. Clustal Omega also converts the initial alignment into an HMM that is then aligned in the background to the individual sequences and subprofiles so that Clustal Omega can "anticipate" how and where the other sequences will align to it. The actual method for "anticipating" is to transfer pseudocount information from the HMM of the initial alignment to the sequences and subalignments that have to be realigned; this process is described in greater detail in Sievers et al. (2011). Sequence alignment is particularly vulnerable to misalignment during the early stages of progressive alignment, and the pseudocount transfer to individual sequences and small subalignments can, therefore, be large. As the subalignments grow during the latter stages of progressive alignment, enough "real" information should have accumulated so that the pseudocount transfer can be scaled back. For subprofiles of 100 or more sequences, there is effectively no pseudocount transfer. Alignments can be refined an indefinite number of times; however, experience has shown that one or two iterations produce a good improvement in alignment quality. More than two iterations are rarely useful and should be applied on a case-by-case basis. The minimum command for performing an iterative alignment is written as follows:

```
clustalo -i infile.fa -o outfile1.fa --iter=1
```

where `infile.fa` and `outfile1.fa` are the names of the FASTA-formatted input and output files, respectively.

Keep in mind that the use of iteration comes with a performance penalty. For each round of iteration, three additional alignments have to be performed: the first and the second subalignments have to be aligned with the background HMM, and then two subalignments, augmented with pseudocount background information, have to be aligned themselves. An alignment using one round of iteration takes roughly four times as long as the initial alignment, and an alignment using two rounds of iteration should take roughly seven times as long as the original alignment.

During iteration, a preliminary alignment is converted into an HMM, and this HMM is then used to produce a higher quality alignment. HMM information can be generated externally. If the type of sequences to be aligned is known, then there may already exist a pre-calculated HMM. For example, Pfam (Finn et al. 2016) contains a large collection of protein families, alignments, and their HMMs. If it is known that the sequences to be aligned are homologous to a family in Pfam, then the corresponding HMM can be downloaded from Pfam and used as an additional command-line argument:

```
clustalo -i infile.fa -o outfile4.fa --hmm-in=pfam.hmm
```

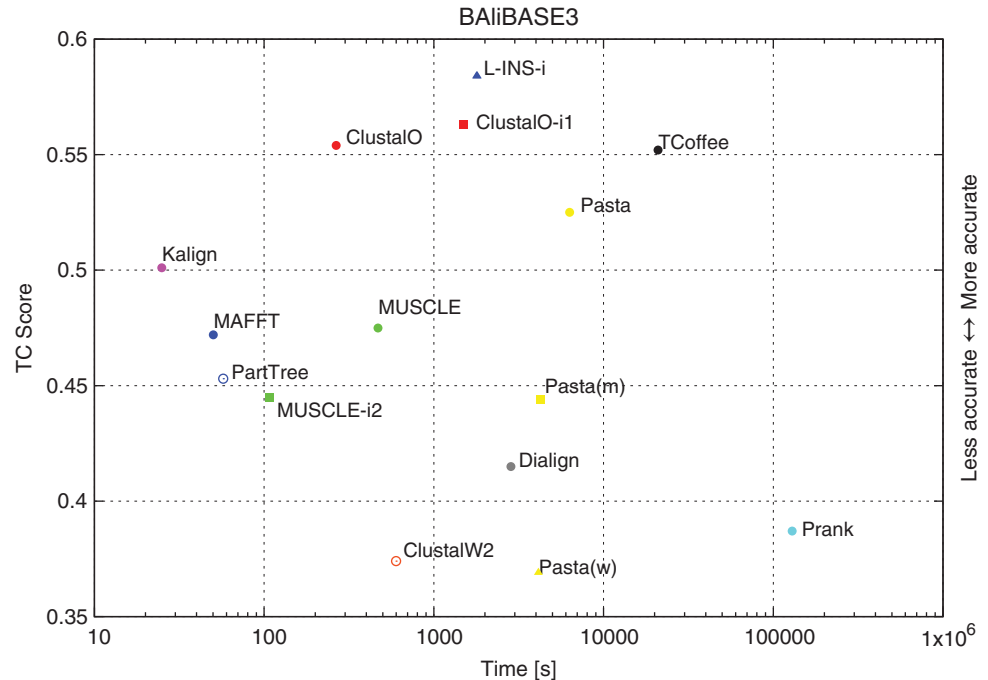


Figure 8.3 Aligner accuracy versus total single-threaded run time using the BALiBASE3 benchmark. Times are sums of and total column (TC) scores are averages over all 218 test alignments. X-axis (time) is logarithmic, while the y-axis (TC Score) is linear. Data points are for aligner default settings. Additional points are for Clustal Omega (i1: more accurate mode), MUSCLE (i2: fast mode), and PASTA (m, MUSCLE as subaligner; w, ClustalW2 as subaligner). Data points correspond to columns 8 and 9 in Table 8.1.

where `pfam.hmm` is an HMM downloaded from Pfam that contains alignment information from a protein family homologous to the sequences contained in `infile.fa`. Alternatively, the HMM could be generated from a locally produced alignment using HMMER (Finn et al. 2011).

Benchmarking Clustal Omega When assessing the performance of a multiple sequence aligner, several issues should be considered. Can the alignment software handle the number of input sequences? How long does the alignment process take? Can the alignment be extended to larger numbers of sequences, or to longer sequences? How accurate are the alignments, when compared with standard alignments of sequences of known three-dimensional structure? Different aligners perform differently in all of these respects. Some are very fast with small sets of sequences but take impractical amounts of time when the number of sequences grows beyond a few hundred. Some of these slow aligners can, however, be very accurate when tested on benchmarks. In contrast, some aligners can handle extremely large datasets, but at the expense of some accuracy. In this section, Clustal Omega is compared with some widely used alignment packages with regard to computer time and alignment accuracy and also the ability to handle long or many sequences. Detailed descriptions of the alignment packages and how to use them are given in the following sections. In Figure 8.3 and Table 8.1, results are shown using the well-established BALiBASE3 benchmark (Thompson et al. 2005). Here, accuracy is measured as the proportion of alignment columns in 218 benchmark alignments and is given as the TC score in the table. Clustal Omega is neither the fastest nor the most accurate alignment package, but it is more accurate than all of the faster aligners, including L-INS-i from the MAFFT package (Kato et al. 2005a,b), the only aligner that achieves a higher TC score (Figure 8.3). Figure 8.3 gives total times and overall accuracy scores for BALiBASE3. BALiBASE3 is divided into subcategories of alignment types and the individual results for these are given in Table 8.1.

The performance measures in Table 8.1 are for a dataset of fixed size. Figure 8.4 plots the run times of various MSA algorithms against the number of sequences to be aligned

Table 8.1 Aligner performance on BALiBASE3 benchmark.

Aligner	BALiBASE reference set							Time	Memory	
	BB11	BB12	BB2	BB3	BB4	BB5	all		RSS	ss
ClustalO	0.36	0.79	0.45	0.58	0.58	0.53	0.55	00 h:04 m:25 s	959 060	55 961
ClustalO-i1	0.36	0.79	0.45	0.59	0.59	0.55	0.56	00 h:24 m:53 s	3 442 156	106 888
ClustalW2	0.22	0.71	0.22	0.27	0.40	0.31	0.37	00 h:09 m:58 s	8 032	3 852
DIALIGN	0.27	0.70	0.29	0.31	0.44	0.43	0.42	00 h:47 m:28 s	56 912	7 350
Kalign	0.37	0.79	0.36	0.48	0.50	0.44	0.50	00 h:00 m:24 s	7 260	2 776
L-INS-i	0.40	0.84	0.46	0.59	0.60	0.59	0.58	00 h:30 m:01 s	703 524	43 695
MAFFT	0.29	0.77	0.33	0.42	0.49	0.50	0.47	00 h:00 m:50s	461 668	35 950
PartTree	0.28	0.76	0.30	0.40	0.45	0.50	0.45	00 h:00 m:57 s	448 524	19 421
MUSCLE	0.32	0.80	0.35	0.41	0.45	0.46	0.48	00 h:07 m:48 s	78 608	15 892
MUSCLE-i2	0.27	0.76	0.33	0.38	0.43	0.43	0.45	00 h:01 m:47 s	78 780	15 860
PASTA(w)	0.24	0.71	0.23	0.23	0.37	0.34	0.37	01 h:08 m:49 s	317 112	58 703
PASTA	0.35	0.78	0.45	0.50	0.51	0.52	0.53	01 h:45 m:08 s	664 336	65 448
PASTA(m)	0.30	0.78	0.31	0.35	0.44	0.39	0.44	01 h:10 m:43 s	323 936	62 038
PRANK	0.24	0.68	0.25	0.35	0.36	0.39	0.39	35 h:55 m:53 s	468 692	36 742
T-Coffee	0.41	0.86	0.40	0.47	0.55	0.59	0.55	05 h:48 m:46 s	1 870 536	192 504
	38	44	41	30	49	16	218			

Columns 2–7 (BB11–B5) average total column (TC) scores for hierarchical reference sets; column 8 (all) TC scores averaged over all 218 test alignments. Column 9 (time) total (single threaded) run time for all 218 test alignments. Column 10 (RSS) maximum memory requirements; column 11 (rss) average memory requirement. Columns 8/9 (all/time) are represented in Figure 8.4. Last row gives numbers of test alignments in each hierarchical set.

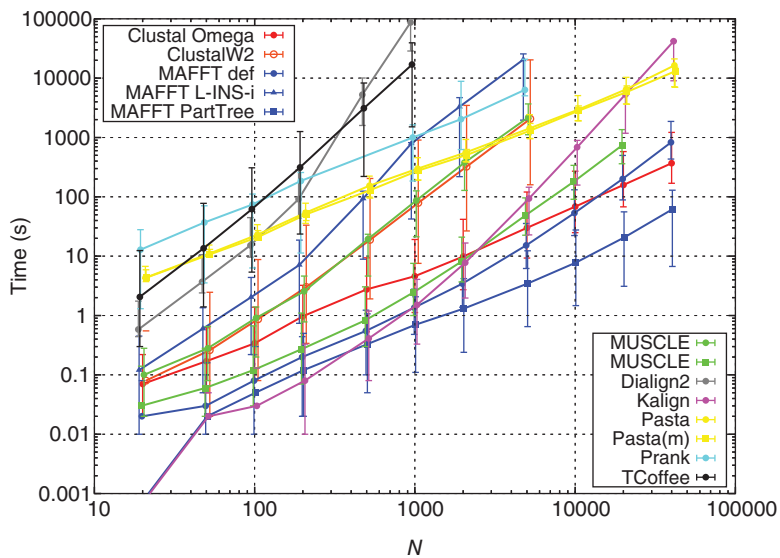


Figure 8.4 Total single-threaded execution time (y-axis) for different aligners as the number of sequences (x-axis) is increased. Both axes are logarithmic. Whiskers give the range of times from short sequences (lower whisker: zf-CCHH/PF00096, length from 23 to 34 residues) to long sequences (upper whisker: RuBisCO_large/PF00016, length from 295 to 329). Solid lines connect times for median-length sequences (rvp/PF00077, length from 94 to 124).

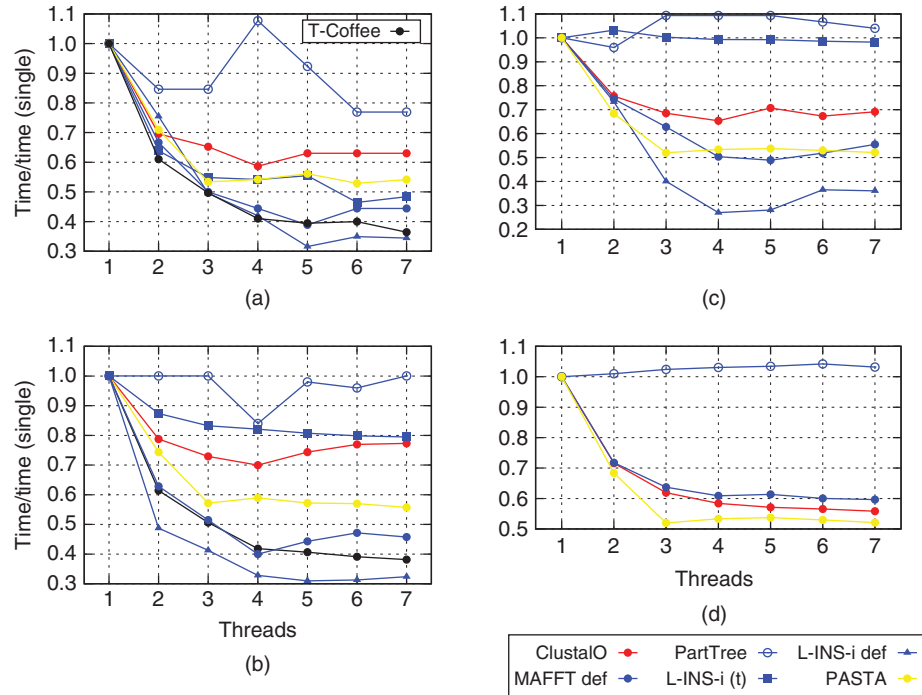


Figure 8.5 Ratio of total run time relative to single-threaded execution (y-axis) as different numbers of threads (x-axis) are employed: (a) for 100 sequences; (b) for 500 sequences; (c) for 1000 sequences; and (d) for 10 000 sequences. Def refers to default settings for the program. L-INS-i (t) refers to the non-threaded setting for that program.

for three sets of sequences of different lengths, taken from Pfam (Finn et al. 2014). The bars correspond to results for a very short protein domain (zf-CCHH, average length 23 amino acids), a medium-length domain (rvp, average sequence length 93), and a long protein domain (RuBisCO_large, length 248). The results in Figure 8.4 are represented as a double-logarithmic plot. Shallow curves scale favorably; that is, an increase in the number of sequences being aligned will only result in a moderate increase in calculation time. Steep curves scale unfavorably, with calculation times increasing rapidly when using larger and larger sequence sets. Results for Clustal Omega are represented by red bars (zf-CCHH at the bottom, RuBisCO_large at the top) and bullets (rvp). For datasets of 20–1000 sequences, Clustal Omega is slower than Kalign (magenta bullets), default MAFFT (dark blue bullets), or fast MUSCLE (green squares). Owing to its more favorable scalability, Clustal Omega overtakes fast MUSCLE and Kalign at $N = 2000$ and default MAFFT at $N = 20\,000$. MAFFT PartTree (dark blue squares) is consistently faster than Clustal Omega over all datasets.

Both main stages of the progressive alignment heuristic (that is, the distance calculation and the pairwise alignment) have been parallelized in Clustal Omega. An alignment may be distributed among the different cores of one computer but not among different computers. Distance matrix calculation is an easily parallelizable task. The pairwise alignment stage, on the other hand, is difficult to parallelize effectively. As can be seen in Figure 8.5, good speed-up for Clustal Omega is attainable for two, three or four threads, but more threads are only useful if the number of sequences is very large. Parallelization in Clustal Omega is “thread-safe,” with an alignment generated using one thread guaranteed to be the same as when using more than one thread.

ClustalW2

ClustalW2 (Larkin et al. 2007) is the predecessor of Clustal Omega and is derived from a series of programs dating back to the 1980s. It is usually slower than Clustal Omega, it cannot align as many sequences, and its alignments are often of lower quality. It also cannot be sped up by using

multiple threads. Since 2010, its code base has been frozen, and ClustalW2 is no longer under active development. While ClustalW2 is still available as an online tool at the Pasteur Galaxy server, it is no longer provided as a tool by the EBI or the Max Planck Bioinformatics Toolkit. ClustalW2 is part of several Linux distributions (e.g. Ubuntu, with code and executables available from the Clustal web site). ClustalW2 is included here because it is still very widely used and the GUI makes it a very easy and intuitive program to use. Unlike Clustal Omega, it can be run interactively in a terminal or with a GUI, known as ClustalX. Here, however, we will only describe the use of ClustalW2 from the command line.

ClustalW2 is also a progressive aligner and always calculates a full $N \times N$ distance matrix, where N is the number of sequences to be aligned. This effectively limits the number of sequences that can be aligned by ClustalW2 in a reasonable amount of time. Here, we did not attempt to align more than 5000 sequences. The “W” in the name ClustalW derives from the weighting scheme for down-weighting over-represented sequences.

ClustalW2 automatically recognizes seven sequence file formats as input: NBRF-PIR, EMBL-SWISSPROT, Pearson (FASTA), Clustal, GCG-MSF, GCG9-RSF, and GDE. Alignment output is by default in Clustal format but GCG, NBRF-PIR, PHYLIP, GDE, NEXUS, or FASTA can be selected. A minimal ClustalW2 command line would be written as follows:

```
clustalw2 -INFILE=infile.fa
```

This will read the sequences in `infile.fa`, detect the file format, guess whether the sequences are nucleotide or protein, align the sequences, and write the alignment in Clustal format into a file `infile.aln`. The stem of the input file name (in this example `'infile'`) is retained, the file extension (in this example `'.fa'`) is dropped and the extension `'.aln'` is appended. ClustalW2 also outputs by default the guide tree in Newick format to a file that ends with `'.dnd'`. A progress report, specifying distances of unaligned sequences and intermediate subalignment scores, is printed to standard output. For large numbers of sequences, this can be time and memory consuming and may be suppressed by setting the `-QUIET` flag. If the alignment should be written to a file with a name different than the default name, then this can be specified by setting the `-OUTFILE` flag. The output format can be specified by setting the `-OUTPUT` flag, as shown in the following command:

```
clustalw2 -INFILE=infile.fa -OUTFILE=output.a2m -OUTPUT=fasta
```

Here, the alignment of unaligned sequences in `infile.fa` will be written to a file `output.a2m` in FASTA format.

On the standard protein benchmark BALiBASE3, ClustalW2 exhibits a medium speed that is slower than Clustal Omega, default MAFFT, and Kalign. Its execution time is roughly the same as default MUSCLE but it is faster than PRANK, T-Coffee, and PASTA. However, its accuracy, as measured by the TC score, is the worst of all the aligners considered here, as can be seen in Figure 8.3.

ClustalW2 is, for small numbers of sequences, one of the most memory frugal aligners in this comparison. Its time and memory requirements, however, grow with the square of the number of sequences. Therefore, we could not extend the range of sequence numbers beyond 5000 on our benchmark machine with 8GB of RAM, as can be seen from the orange circles in Figure 8.4.

DIALIGN

The progressive alignment algorithm is appropriate if the sequences to be aligned are clearly alignable over their full lengths. However, if sequences only share local similarities and are otherwise unrelated, then this may not be suitable. For example, sequences which share one short protein domain but which are otherwise completely unrelated will be difficult to align with the standard progressive aligners such as Clustal Omega. DIALIGN (Morgenstern et al. 1998) does not attempt to match up individual residues but rather segments of residues. These segments are free from gaps and have the same lengths in all sequences to be aligned. While

there are no gaps within the segments, mismatches are allowed. Segments of different lengths are considered, but a lower threshold of 10 is normally used. Multiple segments are aligned if this can be done consistently – that is, where no segment is aligned to more than one segment in another sequence and where all segments in all sequences are in the same order. This consistency scheme pre-dates the one implemented in T-Coffee (Notredame et al. 2000). A typical command line for DIALIGN is

```
dialign2 -fa input.in
```

where `input.in` is the file that contains the unaligned sequences to be aligned. Aligned output is written to a file with the same name as the input file and an added extension `.fa'`. Using the BALiBASE3 benchmark dataset, DIALIGN is faster than T-Coffee, PASTA, and PRANK but slower than all the other aligners. TC scores for DIALIGN are relatively low, but they outperform ClustalW2 and PRANK. DIALIGN's run time requirements are the highest for all the aligners (Figure 8.4). DIALIGN's memory requirements are initially low, but appear to grow quadratically with the number of sequences (Figure 8.5). A version of DIALIGN has been parallelized (Schmollinger et al. 2004).

Kalign

Kalign2 (Lassmann and Sonnhammer 2005) is a progressive MSA program. It establishes the distances necessary for generating the guide tree by using the Muth–Manber string-matching algorithm (Muth and Manber 1996). This appears to be the fastest distance calculation algorithm of all the programs considered here. Distance matrix calculation in Kalign2, however, scales quadratically with the number of sequences. Kalign2 offers support for Clustal, PileUp, MSF, Stockholm, UniProt, Swiss-Prot, and Maccim alignment formats.

A minimum Kalign2 command line is written as follows:

```
kalign -in input.fa -out output.fa
```

This command will write the alignment of unaligned sequences in `input.fa` into `output.fa` in default FASTA format. Additionally, a progress report is written to standard output.

Using the BALiBASE3 benchmark, Kalign2 is by far the fastest of the programs considered here. Its accuracy, as measured by the TC score, is better than the default versions of MAFFT, MUSCLE, or ClustalW2, but not as high as L-INS-i, Clustal Omega, or T-Coffee (Figure 8.3). With between four and 142 sequences (median 21 sequences), however, BALiBASE3 is a relatively small benchmark. For larger sequence numbers, Kalign's scalability outweighs its efficient implementation and is overtaken in terms of speed by MAFFT (for 1000 sequences), MUSCLE in fast mode, Clustal Omega (for 2000 sequences), and PASTA (for 20 000 sequences).

MAFFT

MAFFT (Katoh et al. 2005a,b) is a collection of different executables, managed by a script that selects a range of multiple aligners depending on the number of sequences, the desired accuracy, and available computational power. Here we will focus on (i) the general purpose default MAFFT aligner FFT-NS-i for medium to large datasets, (ii) the more accurate but slower L-INS-i for small datasets of a few hundred sequences, and (iii) PartTree, which can deal with overwhelmingly large numbers of sequences.

When MAFFT is run without specifying a particular aligner, it runs in default mode. In default mode, MAFFT recodes an amino acid sequence as a sequence of tuples, comprising the residues' volume and polarity. The correlation of the volumes and polarities of two sequences can be efficiently calculated, using a fast Fourier transform (FFT). This way, homologous sections of the sequences are identified. These parts are then aligned using conventional dynamic programming. This algorithm is referred to as FFT-NS-1. In default mode, MAFFT repeats this process one more time (referred to as FFT-NS-2) and then employs an iterative refinement, which finally constitutes FFT-NS-i. The MSA produced during FFT-NS-2

is gradually refined by repeated pairwise alignment of randomly partitioned groups in the sequences. L-INS-i uses iterative refinement, as well as alignment consistency (Notredame et al. 2000), a technique that measures consistency between the multiple alignment and pairwise alignments. This approach can be very accurate but, in general, scales cubically with the number of sequences. It is, therefore, mostly applicable to smaller problems. On the other hand, PartTree is a fast method that quickly constructs a guide tree, allowing it to handle datasets of many thousands of sequences.

Default MAFFT A minimum default MAFFT command line is written as follows:

```
mafft input.fa > output.fa
```

MAFFT does not accept non-standard amino acid symbols, such as ambiguity codes. If any such symbols are among the sequence information, then the `--anysymbol` flag should be set. Diagnostic output (to standard error) can be suppressed by setting the `--quiet` flag.

On BALiBASE3, default MAFFT is the second fastest aligner after Kalign2, with a TC score slightly below Kalign2, comparable to default MUSCLE, and much higher than ClustalW2. Memory consumption is consistently high. All MAFFT strategies have been parallelized, and speed-up is good for up to four threads. Beyond that, useful speed-up is achieved only for very large numbers of sequences. Default MAFFT is thread-safe, i.e. an alignment generated using one thread is guaranteed to be the same as when using more than one thread. This means that alignments are repeatable in multi-thread mode.

L-INS-i L-INS-i is the high-accuracy MAFFT program and, consequently, has lower throughput than the default version. A minimum MAFFT L-INS-i command line can be written in one of two ways:

```
linsi input.fa > output.fa
```

or

```
mafft --localpair input.fa > output.fa
```

Of all the programs considered here, MAFFT L-INS-i attains the highest TC score on the BALiBASE3 benchmark (Figure 8.3). Its execution time is slower than MUSCLE and Clustal Omega, comparable to one iteration of Clustal Omega, and faster than either T-Coffee or PASTA. The speed-up for multi-threaded execution of MAFFT L-INS-i is the best of all the programs. However, MAFFT L-INS-i is not thread-safe. This means that the results of runs using different numbers of threads can differ. Even results using the same number of threads may differ for different runs.

PartTree The minimal MAFFT PartTree command line is written as follows:

```
mafft --parttree input.fa > output.fa
```

PartTree is the high-throughput MAFFT program and is not expected to do well on a small benchmark like BALiBASE3. It is slower and less accurate than the MAFFT default version. The data in Figure 8.4 show that PartTree is consistently the fastest aligner for more than 200 sequences. Clustal Omega has a similar scalability (Figure 8.4) but has a higher overhead. PartTree is also the most memory-efficient algorithm for more than 2000 sequences. Guide trees can be written in all versions of MAFFT by setting the `--treeout` flag. In PartTree, however, the sequence identifiers are replaced by the integer index of the position in which the sequence appears in the input file. PartTree guide trees also may contain multi-furcations. As in all versions of MAFFT, external guide trees may be read in, however, the file format is exclusive to MAFFT. The input has to be generated from a guide tree in standard format using a utility program called `newick2mafft.rb`; this program is part of the MAFFT distribution. PartTree is thread-safe; however, there is no useful speed-up for more than one thread.

MUSCLE

MUSCLE (Edgar 2004) is a progressive MSA program. During a first stage, it calculates a distance matrix of the unaligned sequences, based on a fast k -tuple vector comparison. These distances are then clustered using UPGMA cluster analysis (Sokal and Michener 1958). This stage produces a first alignment, which can then be improved upon in a second iterative step. This step is similar to the first, the only difference being that alignment-based distances (Kimura 1983) are used instead of k -tuple vector comparisons. During a subsequent round of iterative refinements, the second-stage alignment can be improved upon by cutting the second-stage guide tree into two parts, realigning the sequences in each subtree, and aligning the two sub-profiles (called tree-dependent restricted partitioning). The new alignment is accepted if its alignment score is improved. These refinements are, by default, carried out 14 times, leading to overall 16 rounds of alignments.

A minimal MUSCLE command line is written as follows:

```
muscle -in input.fa -out output.fa
```

This command will carry out the initial two rounds of alignment (both k -tuple and alignment distance based), followed by 14 rounds of iterative refinement. If the number of sequences is large, the iterative refinement can be skipped by including an additional term in the command specifying the maximum number of iterations:

```
muscle -in input.fa -out output.fa -maxiters 2
```

Using the BALiBASE3 benchmark, default MUSCLE has an accuracy (as measured by the TC score) that is comparable to that of default MAFFT; it is slightly faster than ClustalW2 and slightly slower than Clustal Omega. Fast MUSCLE, with only the first two alignment phases, is roughly one order of magnitude faster than its default version. Using BALiBASE3, it is faster than Clustal Omega but still not as fast as default MAFFT or Kalign2. Its accuracy, however, falls off with respect to the default version. In the large-scale tests in Figure 8.4 MUSCLE exceeded the memory available in our test bed for 5000 and 20 000 sequences for default and fast mode, respectively. The run times for fast MUSCLE start off faster than Clustal Omega and slower than Kalign2 but overtake Kalign2 in terms of speed and are, in turn, overtaken by Clustal Omega for 2000 sequences. As the iterative refinement re-partitions the guide tree but does not regenerate it, guide trees are always the same for the default and the fast version. There is no parallel version of MUSCLE.

PASTA

PASTA (Practical Alignments using SATé and TrAnsitivity; Mirarab et al. 2015) is a Python script that calls existing software packages, such as SATé (Liu et al. 2009), MAFFT, MUSCLE, ClustalW, HMMER (Eddy 2009), OPAL (Wheeler and Kececioglu 2007), and FastTree-2 (Price et al. 2010), and combines their results. In a first step, a small number of sequences is randomly selected from the input dataset and aligned. The default aligner for PASTA is MAFFT L-INS-i. This initial alignment is called the “backbone” and is converted into an HMM using HMMER. The remaining sequences are aligned onto this HMM. An initial maximum likelihood (ML) tree is constructed from this alignment using FastTree. The sequences are then clustered according to this tree, such that the cluster size is small. The clusters are then aligned using the default aligner to form subalignments. Subalignments that are “adjacent” in the overall spanning tree are aligned using OPAL to form subalignment pairs. Different subalignment pairs are merged to produce the overall alignments.

PASTA expects nucleotide sequence input, by default. For protein sequences, a minimal PASTA command line would be written as follows:

```
python run_pasta.py --input=input.fa --datatype=Protein
```

Using the BALiBASE3 benchmark, default PASTA is faster than T-Coffee and PRANK but slower than all the other aligners. PASTA’s accuracy intimately reflects the accuracy of the

underlying subalignment software. This aligner can be changed by specifying, for example, `--aligner=muscle` or `--aligner=clustalw2`. PASTA alignments are more accurate if a more accurate aligner such as L-INS-i is used, as in the default. They are of medium quality if an aligner such as MUSCLE is used, and PASTA produces the worst alignments if ClustalW2 is used. For BALiBASE3, PASTA alignments, however, never quite reach or exceed the quality of the underlying subalignment software. This can be seen in Figure 8.3, with the PASTA data points being to the right (slower) and below (less accurate) the points of its corresponding subaligner. This is not surprising, as it has been demonstrated using alignments of small numbers of protein sequences that ML phylogenetic trees are not necessarily good guide trees and frequently are decidedly bad guide trees (Sievers et al. 2014).

PASTA, however, was not designed for aligning small numbers of sequences. Using the large-scale benchmark data, it starts off (at 20 sequences) as the second slowest aligner after PRANK but, because of its favorable time scalability, it overtakes L-INS-i at 500 sequences, default MUSCLE and ClustalW2 at 5000 sequences, and Kalign2 at 20 000 sequences. Its memory consumption scales similarly.

PASTA has been parallelized. By default, it tries to use all available threads. The number of threads can be changed by specifying an argument for the `--num_cpus` flag. PASTA demonstrates a good speed-up as the number of threads is increased; this effect becomes even more pronounced as the number of sequences is increased, as can be seen in Figure 8.5. However, the version of PASTA examined here is not thread-safe. This means that alignments can differ depending on the number of threads. Probably even more disconcertingly, alignments cannot be recreated using more than one thread. This is true in PASTA's default mode, which uses non-thread-safe L-INS-i, and also if MUSCLE, which is single-threaded only, is used as the sub-aligner. For the latter, in one particular example, alignment lengths can vary from 159 to 183 if the same 100 rvp sequences (average length 106.5, longest sequence 124) are aligned 10 times using three threads. In this example, the TC scores of the core columns of the six rvp reference sequences vary between 0.433 and 0.556. Therefore, one should always set `--num_cpus=1` so that results are reproducible.

PRANK

In a pairwise alignment of two single sequences, one cannot decide if a gap in one sequence is caused by a deletion in this sequence or by an insertion in the other sequence. In an MSA, however, such a distinction may become important, especially for phylogenetic analysis. Most progressive aligners underestimate the true number of insertion events and can give rise to artificially short alignments. PRANK (Löytynoja and Goldman 2005) tries to account for this fact by performing a phylogeny-aware gap placement. This makes PRANK potentially useful for sequences where one is interested in carefully estimating the locations of all gaps. It cannot properly be tested by the type of structure-based benchmarks described here, and its low performance does not mean it is not useful in other situations.

A minimum command line for PRANK is written as follows:

```
prank -d=infile.fa -o=outfile -f=fasta
```

Using the BALiBASE3 benchmark, PRANK is the slowest aligner and, with the exception of ClustalW2, the one that attains the lowest TC score (Figure 8.3). This is unsurprising, as conventional, structure-based benchmarks reward compact alignments and possibly do not penalize over-alignment sufficiently. It should be noted that PRANK only reads standard IUPAC codes (unique letters for each amino acid or base) and replaces all non-IUPAC characters (like ambiguity codes) with N or X. Comparing the alignment with the unaligned data or with a reference alignment can, therefore, lead to discrepancies.

The scalability benchmark shows that PRANK is a slow aligner for small numbers of sequences. However, PRANK's time complexity is one of the lowest of all the aligners: after 100 sequences PRANK overtakes T-Coffee, and after 1000 sequences it overtakes MAFFT

L-INS-i (Figure 8.4). Its memory requirements follow a similar trend and are predicted to exceed the available memory of this test bed after 5000 sequences.

T-Coffee

T-Coffee started as a progressive alignment heuristic method for optimizing the Coffee objective function for MSA (Notredame et al. 1998). That function finds the MSA that maximizes the sum of weighted pairwise matches between residues from different sequences. Those pairwise matches can come from pairwise alignments, existing MSAs, corresponding residues from protein structure superpositions, or aligned residues from RNA structure alignments. That makes it possible for T-Coffee to merge alignment information from unaligned sequences, different MSA packages, structure alignments, or a mixture of these. In Notredame et al. (2000), MSA *consistency* was first described, where pairwise residue matches between sequences that agree with pairwise matches from other pairs get increased weight. This helps to get around the inherently greedy nature of progressive alignment and was shown to give very accurate alignments. Consistency has since been incorporated into the Probcons (Do et al. 2005) and MAFFT (Katoh et al. 2005a,b) packages. It adds to the computational complexity of the alignment and is mainly suitable for aligning less than 1000 sequences but it greatly increases alignment accuracy.

A minimum command line for T-Coffee is written as follows:

```
t_coffee -in infile.fa -output fasta
```

This command will produce an alignment file named `infile.fasta_aln` in FASTA format.

Using the BALiBASE3 benchmark, T-Coffee is the second slowest aligner after PRANK. Its average TC score, however, is among the highest, beating PASTA, Kalign, and MUSCLE, as shown in Figure 8.3. T-Coffee's average memory consumption is the highest. As T-Coffee is based on the principle of consistency, its time complexity with respect to the number of sequences is expected to be high. We could not extend the sequence range beyond 1000 as T-Coffee exhausted the available 8GB of RAM. In terms of parallelization, T-Coffee is completely thread-safe. This means that alignments do not depend on the number of processors, which can be set by specifying the `-n_core` flag. Alignments are also reproducible. T-Coffee is, therefore, the aligner with the best parallel speed-up while still being thread-safe.

Viewing a Multiple Alignment

It is very difficult to view an MSA without using visualization software to emphasize some features of the alignment. For example, conserved columns or motifs can be emphasized by using different fonts or colors or shading. Further, an alignment can be annotated by showing structural or functional features in different regions. There are some dedicated alignment viewing packages and packages which include very good viewing capabilities, and we mention some widely used ones below (see Internet Resources). Two of these packages (SeaView and Jalview) also include extremely powerful capabilities for editing an MSA.

Clustal X

Clustal X (Thompson et al. 1997) was created by taking the pre-existing Clustal W package (Thompson et al. 1994) and adding a GUI that was portable across all widely used operating systems. The alignment engine is identical in the two packages, and both were developed and maintained afterwards in parallel. The unaligned or aligned sequences are shown in a scrollable window with a default coloring scheme that emphasizes residues that are well conserved in columns. Clustal X includes tools for manipulating the alignment display by user-adjustable coloring schemes, font sizes, and options for highlighting poorly conserved blocks, columns,

or sequences. Alignments can also be produced as high-quality PostScript files for publication. These coloring facilities work best for amino acid sequences, but nucleotide sequences can also be viewed. Clustal X is no longer actively developed but it is still freely available and widely used owing to its portability, robustness, and ease of use. It is available as a desktop application for all widely used operating systems.

Jalview

The Jalview open-source MSA editor and analysis workbench works on the Windows, Mac, and Linux platforms (Waterhouse et al. 2009). Jalview focuses on multiple alignments and functional analyses at the gene, protein, or RNA family level rather than on whole genomes. In addition to sophisticated interactive multiple alignment editing functions for DNA, RNA, and protein sequences, including “undo,” multiple “views,” and the ability to subset and “hide” sequences and columns of an alignment, Jalview provides linked views of trees, DNA and protein sequences, protein three-dimensional structures via Jmol or Chimera (Pettersen et al. 2004), and RNA secondary structure via VARNA (Darty et al. 2009). Two examples are shown in Figure 8.6: a protein alignment, with linked protein structure displays, and an RNA alignment, linked to an RNA secondary structure display. Jalview connects to major public databases of sequences, alignments, and three-dimensional structures to allow easy access to these resources and sequence annotations (e.g. active site descriptions). Jalview supports a wide range of annotation methods both on individual sequences and calculated from alignment columns to be displayed on or under the alignment. It also includes a split DNA/RNA/protein view that links DNA alignments and the associated protein sequence alignments to be edited and analyzed together; an example is shown in Figure 8.7. This view also permits the mapping of population variation data single-nucleotide polymorphisms (SNPs) and other genomic features such as gene exons to protein sequences and three-dimensional structure. For example, a Jalview user can look up proteins in UniProt, then cross-reference them back to the full gene and transcripts in Ensembl to see any known SNPs on the alignment, then view the three-dimensional protein structures and location of SNPs (if available) with a few clicks of the mouse.

In order to make alignments, Jalview includes direct access to eight popular multiple alignment algorithms and allows users to modify the parameters for each method (Troshin et al. 2011). Thus, users can interactively align, realign, and compare alignments generated by different methods and parameter combinations. Jalview also provides direct access to the JPred protein secondary structure prediction algorithm (Drozdetskiy et al. 2015) to predict protein secondary structure and solvent accessibility from either a single sequence or a multiple alignment. Jalview includes four protein disorder prediction algorithms and the RNAalifold program (Bernhart et al. 2008) that predicts RNA secondary structure from RNA multiple alignments via JABAWS2.2. For conservation analysis, there are 17 different amino acid conservation score methods, as well as the SMERFS functional site prediction algorithm available in Jalview via the AACon package. The Jalview web site includes training materials and manuals while the online training YouTube channel provides more than 20 short video tutorials on basic and advanced features of Jalview.

SeaView

SeaView (Galtier et al. 1996) is an MSA editor that is especially useful for linking views of an alignment to MSA and phylogenetic packages. It works with either nucleotide or amino acid alignments. SeaView reads and writes a large variety of MSA file formats and can directly call MUSCLE or Clustal Omega to create an MSA. Users can then edit the alignment and call the Gblocks filter program to remove poorly aligned regions. The package can calculate phylogenetic trees using various methods including maximum parsimony (using Protpars from the PHYLIP package; Felsenstein 1981), neighbor joining (Saitou and Nei 1987), or ML

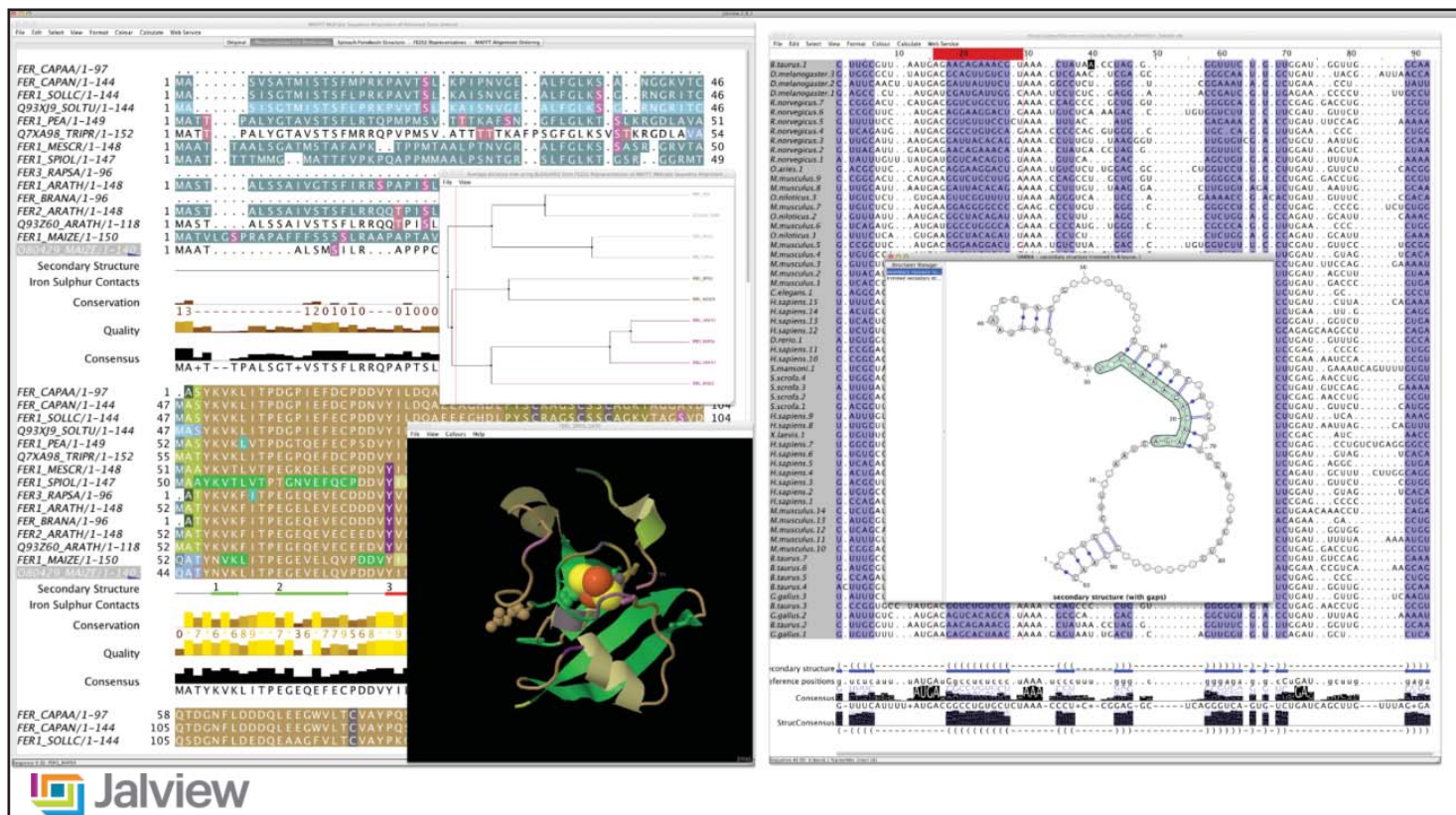


Figure 8.6 Protein and RNA multiple sequence alignments as visualized using Jalview. The left panes illustrate protein multiple alignments with different feature coloring, tree, and Jmol molecular structure views. All windows are linked, so clicking on a residue or sequence in one window will highlight the corresponding residue or sequence in all other windows. On the right, an RNA multiple alignment is illustrated, with corresponding secondary structure information displayed in VARNA.

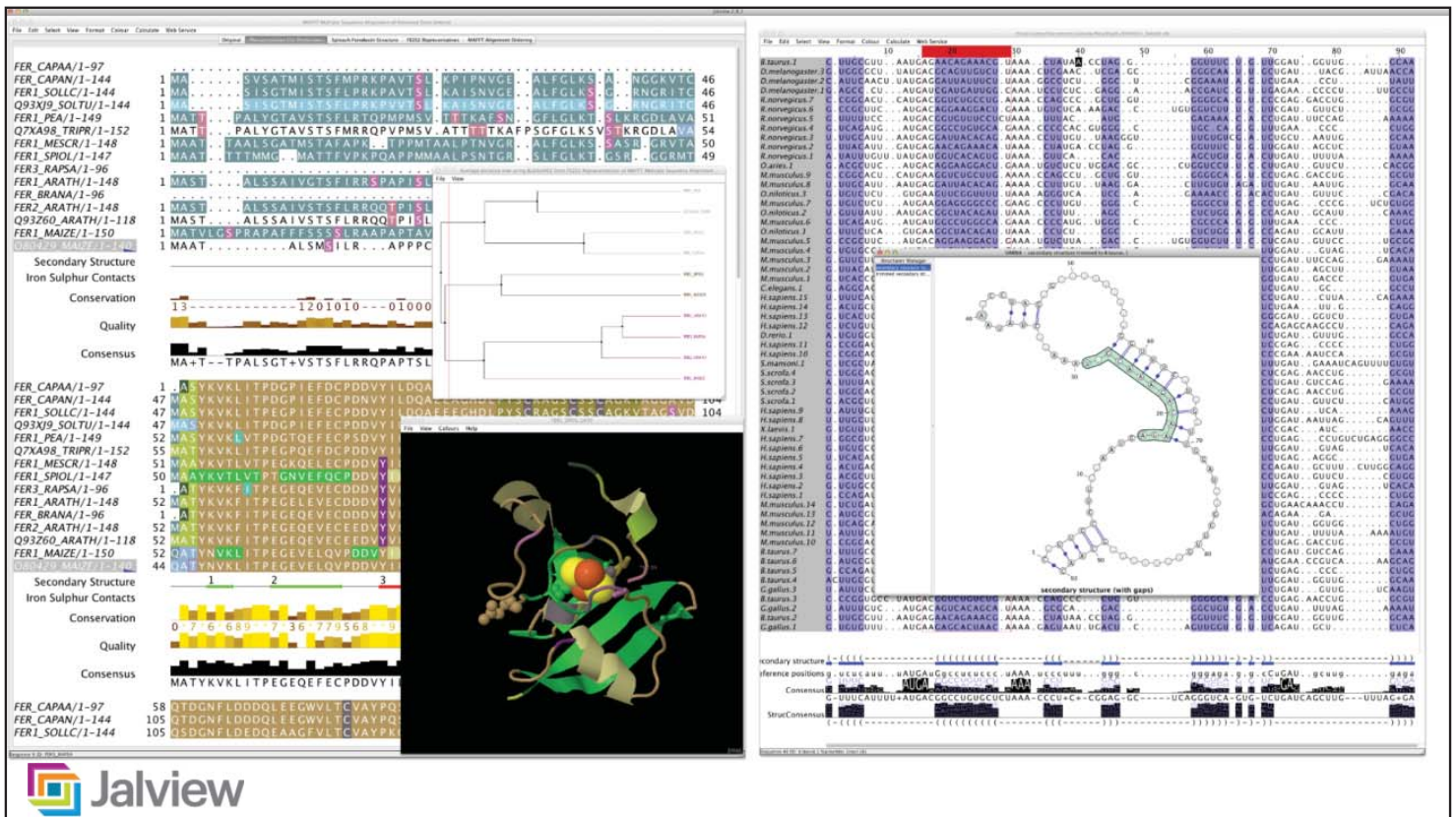


Figure 8.7 Linked coding sequence (CDS), protein, and three-dimensional structure views visualized in Jalview, showing the positions of known single-nucleotide polymorphisms (SNPs). A text search was used in Jalview to find a set of related protein sequences in UniProt. Jalview then cross-referenced these sequences to CDS data found in Ensembl. The protein sequences were multiply aligned by Clustal Omega. Finally, the three-dimensional structure of one of the proteins was displayed in the linked Chimera application. Red and green positions in the alignment highlight the location of known SNPs retrieved from Ensembl.

using Phylml (Guindon and Gascuel 2003). SeaView is a very direct and robust way to go from unaligned sequences to a full phylogenetic analysis under a single framework.

ProViz

ProViz (Jehl et al. 2016) is a recently re-released package for viewing pre-made alignments of protein sequences with superimposed annotation of features, especially functional domains. The alignments and links to databases of functional information are pre-computed, and the viewer displays information about the sequences from a range of sources in an integrated fashion. ProViz can be run online or can be downloaded and run locally. The easiest starting point for viewing is by using the ID, name, or keyword for a protein or gene of interest; the viewer will then show alignments containing that protein. Users can also input their own protein sequence or multiple alignment. The data sources used by ProViz are listed in Internet Resources.

Summary

MSAs of even quite large datasets of thousands of sequences can be carried out quickly online or using Linux-based laptop and desktop computers. These are used in a great variety of further analyses and crop up in almost all phylogenetic and many structural analyses or investigations of sequence similarity. There are many packages available and none of them can be said to give the “best” alignments in all cases; they all use a variety of computational shortcuts, just to make the computations tractable. Different packages have different strengths and weaknesses, and the best solution is to look at the alignments themselves using an alignment viewer and to try different programs out. Some web sites and some alignment-viewing packages support several of the most widely used programs while maintaining a consistent interface. That makes the task of trial and comparison easier. By far, the most important consideration is the nature and quality of the input sequences. They must be sufficiently similar so that they can be aligned; keep in mind that the more fragmentary or outlier sequences that are included, the more fragmented will be the alignment. Clean datasets will give clean alignments that will be easy to view by eye and easy to analyze.

Internet Resources

Multiple sequence alignment versions		
Clustal Omega v1.2.3	www.clustal.org/omega	EMP
ClustalW2 v2.1	www.clustal.org/clustal2	--P
DIALIGN v2.2.2	dialign.gobics.de	---
Kalign v2.04	msa.sbc.su.se/cgi-bin/msa.cgi	---
MAFFT v7.309	mafft.cbrc.jp/alignment/software	EMP
MUSCLE v3.8.31	www.drive5.com/muscle	EMP
PASTA v1.6.4	github.com/smirarab/pasta	---
PRANK v.150803	wasabiapp.org/software/prank	E--
T-Coffee 11.00.8cbe486	www.tcoffee.org/Projects/tcoffee/index.html	EMP

Availability at three sites for online usage (EMP) at EBI (E, www.ebi.ac.uk/services), MPI for Genetics in Tübingen (M, toolkit.tuebingen.mpg.de), and the Pasteur Institute Galaxy server (P, galaxy.pasteur.fr).

Multiple sequence alignment visualization packages

ClustalX	Desktop MSA version of Clustal W	www.clustal.org
Jalview	Alignment editor and viewer	www.jalview.org
SeaView	Alignment editor and viewer	doua.prabi.fr/software/seaview
ProViz	Alignment and annotation viewer	proviz.ucd.ie

Data sources used by ProViz for visualizing protein alignments

Multiple sequence alignments

GeneTree	Homo/Para/ortholog alignments and gene duplication information	www.ensembl.org
GOPHER	Ortholog alignments by reciprocal best hit	bioware.ucd.ie
Quest for orthologs	Datasets of homologous genes	questfororthologs.org

Protein modularity

ELM	Manually curated linear motifs	elm.eu.org
Pfam	Functional regions and binding domains	pfam.xfam.org
Phospho.ELM	Experimentally verified phosphorylation sites	phospho.elm.eu.org

Structural information

DSSP	Secondary structure derived from PDB tertiary structures	swift.cmbi.ru.nl/gv/dssp
Homology models/ SWISS-MODEL	Assigned tertiary structure by sequence similarity to resolved structure	swissmodel.expasy.org
Protein Data Bank (PDB)	Experimentally resolved protein tertiary structures	www.rcsb.org

Genomic data

1000 genomes	Single-nucleotide polymorphism	www.1000genomes.org
dbSNP	Single-nucleotide polymorphism with disease association and genotype information	www.ncbi.nlm.nih.gov/SNP
Isoforms	Alternative splicing	www.uniprot.org

Additional curated data

Mutagenesis	Experimentally validated point mutations and effect	www.uniprot.org
Regions of interest	Experimentally validated functional areas	www.uniprot.org
Switches. ELM	Experimentally validated motif-based molecular switches	switches.elm.eu.org

Prediction

Anchor	Binding sites in disordered regions	anchor.enzim.hu
Conservation	Conservation of residues across the alignment	bioware.ucd.ie
ELM	Linear motifs by regular expression	elm.eu.org
IUPred	Intrinsically disordered regions	iupred.enzim.hu
MobiDB	Collection of various disorder prediction methods	mobidb.bio.unipd.it
PsiPred	Secondary structure for human proteins	bioinf.cs.ucl.ac.uk/psipred

References

- Altschul, S.F., Madden, T.L., Schäffer, A.A. et al. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* 25 (17): 3389–3402.
- Bernhart, S.H., Hofacker, I.L., Will, S. et al. (2008). RNAalifold: improved consensus structure prediction for RNA alignments. *BMC Bioinf.* 9: 474.
- Blackshields, G., Sievers, F., Shi, W. et al. (2010). Sequence embedding for fast construction of guide trees for multiple sequence alignment. *Algorithms Mol. Biol.* 14 (5): 21. <https://doi.org/10.1186/1748-7188-5-21>.
- Chatzou, M., Magis, C., Chang, J.M. et al. (2016). Multiple sequence alignment modeling: methods and applications. *Brief. Bioinform.* 17 (6): 1009–1023.
- Cline, M., Hughey, R., and Karplus, K. (2002). Predicting reliable regions in protein sequence alignments. *Bioinformatics.* 18 (2): 306–314.
- Darty, K., Denise, A., and Ponty, Y. (2009). VARNA: interactive drawing and editing of the RNA secondary structure. *Bioinformatics* 25 (15): 1974–1975.
- Do, C.B., Mahabhashyam, M.S., Brudno, M., and Batzoglou, S. (2005). ProbCons: probabilistic consistency-based multiple sequence alignment. *Genome Res.* 15 (2): 330–340.
- Drozdetskiy, A., Cole, C., Procter, J., and Barton, G.J. (2015). JPred4: a protein secondary structure prediction server. *Nucleic Acids Res.* 43 (W1): W389–W394. <https://doi.org/10.1093/nar/gkv332>.
- Eddy, S.R. (2009). A new generation of homology search tools based on probabilistic inference. *Genome Inf.* 23 (1): 205–211.
- Edgar, R.C. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.* 32 (5): 1792–1797.
- Edgar, R.C. (2010). Quality measures for protein alignment benchmarks. *Nucleic Acids Res.* 38 (7): 2145–2153. <https://doi.org/10.1093/nar/gkp1196>.
- Felsenstein, J. (1981). Evolutionary trees from DNA sequences: a maximum likelihood approach. *J. Mol. Evol.* 17 (6): 368–376.
- Feng, D.F. and Doolittle, R.F. (1987). Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol.* 25 (4): 351–360.
- Finn, R.D., Clements, J., and Eddy, S.R. (2011). HMMER web server: interactive sequence similarity searching. *Nucleic Acids Res.* 39 (Web Server issue): W29–W37. <https://doi.org/10.1093/nar/gkr367>.
- Finn, R.D., Bateman, A., Clements, J. et al. (2014). Pfam: the protein families database. *Nucleic Acids Res.* 42 (Database issue): D222–D230. <https://doi.org/10.1093/nar/gkt1223>.
- Finn, R.D., Coghill, P., Eberhardt, R.Y. et al. (2016). The Pfam protein families database: towards a more sustainable future. *Nucleic Acids Res.* 44 (D1): D279–D285. <https://doi.org/10.1093/nar/gkv1344>.
- Fox, G., Sievers, F., and Higgins, D.G. (2016). Using de novo protein structure predictions to measure the quality of very large multiple sequence alignments. *Bioinformatics.* 32 (6): 814–820. <https://doi.org/10.1093/bioinformatics/btv592>.
- Galtier, N., Gouy, M., and Gautier, C. (1996). SEAVIEW and PHYLO_WIN: two graphic tools for sequence alignment and molecular phylogeny. *Comput. Appl. Biosci.* 12 (6): 543–548.
- Gouy, M., Guindon, S., and Gascuel, O. (2010). SeaView version 4: a multiplatform graphical user interface for sequence alignment and phylogenetic tree building. *Mol. Biol. Evol.* 27 (2): 221–224. <https://doi.org/10.1093/molbev/msp259>.
- Guindon, S. and Gascuel, O. (2003). A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst. Biol.* 52 (5): 696–704.
- Henikoff, S. and Henikoff, J.G. (1992). Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA.* 89 (22): 10915–10919.
- Higgins, D.G., Bleasby, A.J., and Fuchs, R. (1992). CLUSTAL V: improved software for multiple sequence alignment. *Comput. Appl. Biosci.* 8 (2): 189–191.
- Hogeweg, P. and Hesper, B. (1984). The alignment of sets of sequences and the construction of phyletic trees: an integrated method. *J. Mol. Evol.* 20 (2): 175–186.

- Iantorno, S., Gori, K., Goldman, N. et al. (2014). Who watches the watchmen? An appraisal of benchmarks for multiple sequence alignment. *Methods Mol. Biol.* 1079: 59–73. https://doi.org/10.1007/978-1-62703-646-7_4.
- Jehl, P., Manguy, J., Shields, D.C. et al. (2016). ProViz-a web-based visualization tool to investigate the functional and evolutionary features of protein sequences. *Nucleic Acids Res.* 44 (W1): W11–W15. <https://doi.org/10.1093/nar/gkw265>.
- Katoh, K., Kuma, K., Miyata, T., and Toh, H. (2005a). Improvement in the accuracy of multiple sequence alignment program MAFFT. *Genome Inf.* 16 (1): 22–33.
- Katoh, K., Kuma, K., Toh, H., and Miyata, T. (2005b). MAFFT version 5: improvement in accuracy of multiple sequence alignment. *Nucleic Acids Res.* 33 (2): 511–518.
- Kimura, M. (1983). *The Neutral Theory of Molecular Evolution*, 75. Cambridge, UK: Cambridge University Press.
- Konagurthu, A.S., Whisstock, J.C., Stuckey, P.J., and Lesk, A.M. (2006). MUSTANG: a multiple structural alignment algorithm. *Proteins* 64 (3): 559–574.
- Larkin, M.A., Blackshields, G., Brown, N.P. et al. (2007). Clustal W and Clustal X version 2.0. *Bioinformatics.* 23 (21): 2947–2948.
- Lassmann, T. and Sonnhammer, E.L. (2005). Kalign – an accurate and fast multiple sequence alignment algorithm. *BMC Bioinf.* 6: 298.
- Le, Q., Sievers, F., and Higgins, D.G. (2017). Protein multiple sequence alignment benchmarking through secondary structure prediction. *Bioinformatics.* 33 (9): 1331–1337. <https://doi.org/10.1093/bioinformatics/btw840>.
- Liu, K., Raghavan, S., Nelesen, S. et al. (2009). Rapid and accurate large-scale coestimation of sequence alignments and phylogenetic trees. *Science.* 324 (5934): 1561–1564. <https://doi.org/10.1126/science.1171243>.
- Löytynoja, A. and Goldman, N. (2005). An algorithm for progressive multiple alignment of sequences with insertions. *Proc. Natl. Acad. Sci. USA.* 102 (30): 10557–10562.
- Marks, D.S., Colwell, L.J., Sheridan, R. et al. (2011). Protein 3D structure computed from evolutionary sequence variation. *PLoS One.* 6 (12): e28766. <https://doi.org/10.1371/journal.pone.0028766>.
- Mirarab, S., Nguyen, N., Guo, S. et al. (2015). PASTA: ultra-large multiple sequence alignment for nucleotide and amino-acid sequences. *J. Comput. Biol.* 22 (5): 377–386. <https://doi.org/10.1089/cmb.2014.0156>.
- Mizuguchi, K., Deane, C.M., Blundell, T.L., and Overington, J.P. (1998). HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Sci.* 7 (11): 2469–2471.
- Morgenstern, B., Frech, K., Dress, A., and Werner, T. (1998). DIALIGN: finding local similarities by multiple sequence alignment. *Bioinformatics* 14 (3): 290–294.
- Muth, R. and Manber, U. (1996). Approximate multiple string search. In: *Proceedings of the 7th Annual Symposium on Combinatorial Pattern Matching, Laguna Beach, CA (10–12 June 1996)*, vol. 1075, 75–86. Berlin, Germany: Springer.
- Needleman, S.B. and Wunsch, C.D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* 48 (3): 443–453.
- Notredame, C., Higgins, D.G., and Heringa, J. (2000). T-Coffee: a novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol.* 302 (1): 205–217.
- Notredame, C., Holm, L., and Higgins, D.G. (1998). COFFEE: an objective function for multiple sequence alignments. *Bioinformatics.* 14 (5): 407–422.
- Pettersen, E.F., Goddard, T.D., Huang, C.C. et al. (2004). UCSF Chimera: a visualization system for exploratory research and analysis. *J. Comput. Chem.* 25 (13): 1605–1612.
- Price, M.N., Dehal, P.S., and Arkin, A.P. (2010). FastTree 2 – approximately maximum-likelihood trees for large alignments. *PLoS One.* 5 (3): e9490. <https://doi.org/10.1371/journal.pone.0009490>.
- Raghava, G.P., Searle, S.M., Audley, P.C. et al. (2003). OXBench: a benchmark for evaluation of protein multiple sequence alignment accuracy. *BMC Bioinf.* 4: 47.

- Russell, R.B. and Barton, G.J. (1992, 1992). Multiple protein sequence alignment from tertiary structure comparison: assignment of global and residue confidence levels. *Proteins*. 14 (2): 309–323.
- Saitou, N. and Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.* 4 (4): 406–425.
- Sankoff, D., Morel, C., and Cedergren, R.J. (1973). Evolution of 5S rRNA and the non-randomness of base replacement. *Nature*. 245: 232–234.
- Sauder, J.M., Arthur, J.W., and Dunbrack, R.L. Jr., (2000). Large-scale comparison of protein sequence alignment algorithms with structure alignments. *Proteins*. 40 (1): 6–22.
- Schmollinger, M., Nieselt, K., Kaufmann, M., and Morgenstern, B. (2004). DIALIGN P: fast pair-wise and multiple sequence alignment using parallel processors. *BMC Bioinf.* 5: 128.
- Sievers, F., Wilm, A., Dineen, D. et al. (2011). Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal omega. *Mol. Syst. Biol.* 7: 539. <https://doi.org/10.1038/msb.2011.75>.
- Sievers, F., Dineen, D., Wilm, A., and Higgins, D.G. (2013). Making automated multiple alignments of very large numbers of protein sequences. *Bioinformatics*. 29 (8): 989–995. <https://doi.org/10.1093/bioinformatics/btt093>.
- Sievers, F., Hughes, G.M., and Higgins, D.G. (2014). Systematic exploration of guide-tree topology effects for small protein alignments. *BMC Bioinf.* 15: 338. <https://doi.org/10.1186/1471-2105-15-338>.
- Söding, J. (2005). Protein homology detection by HMM-HMM comparison. *Bioinformatics*. 21 (7): 951–960.
- Sokal, R. and Michener, C. (1958). A statistical method for evaluating systematic relationships. *Univ. Kans. Sci. Bull.* 38: 1409–1438.
- Taylor, W.R. and Orengo, C.A. (1989). Protein structure alignment. *J. Mol. Biol.* 208 (1): 1–22.
- Thompson, J.D., Higgins, D.G., and Gibson, T.J. (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 (22): 4673–4680.
- Thompson, J.D., Gibson, T.J., Plewniak, F. et al. (1997). The CLUSTAL_X windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. *Nucleic Acids Res.* 25 (24): 4876–4882.
- Thompson, J.D., Plewniak, F., and Poch, O. (1999). A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Res.* 27 (13): 2682–2690.
- Thompson, J.D., Koehl, P., Ripp, R., and Poch, O. (2005). BALiBASE 3.0: latest developments of the multiple sequence alignment benchmark. *Proteins*. 61 (1): 127–136.
- Troshin, P.V., Procter, J.B., and Barton, G.J. (2011). Java bioinformatics analysis web services for multiple sequence alignment – JABAWS:MSA. *Bioinformatics* 27 (14): 2001–2002.
- Van Walle, I., Lasters, I., and Wyns, L. (2005). SABmark – a benchmark for sequence alignment that covers the entire known fold space. *Bioinformatics*. 21 (7): 1267–1268.
- Waterhouse, A.M., Procter, J.B., Martin, D.M. et al. (2009). Jalview version 2 – a multiple sequence alignment editor and analysis workbench. *Bioinformatics*. 25 (9): 1189–1191. <https://doi.org/10.1093/bioinformatics/btp033>.
- Wheeler, T.J. and Kececioglu, J.D. (2007). Multiple alignment by aligning alignments. *Bioinformatics*. 23 (13): i559–i568.

9

Molecular Evolution and Phylogenetic Analysis

Emma J. Griffiths and Fiona S.L. Brinkman

Introduction

Nothing in biology makes sense except in the light of evolution.

Theodosius Dobzhansky

The universe has been around for a long time. Calculations estimate it to be over 13 billion years old (Planck Collaboration 2015). The solar system is thought to be ~4.6 billion years old (Bouvier and Wadhwa 2010), with the formation of the Earth occurring just slightly later, at ~4.5 billion years ago (Wilde et al. 2001). The earliest evidence for life on Earth has long been considered to be fossilized microbial mats called stromatolites discovered in Western Australia, which date back to 3.4 billion years ago (Wacey et al. 2011). However, recent evidence of biogenic carbon in hydrothermal vent precipitates in Canada date the origins of life as far back as 4.1 billion years (Dodd et al. 2017). That is a long time for organisms to be living, reproducing, interacting, and competing for resources – and, inevitably, dying. And in that time, the Earth has seen many climatic, atmospheric, and geological changes, altering the Earth's chemistry and temperature (Allegre and Schneider 2005).

Because of the similarities in metabolism, physiology, and architecture of cellular life, it is known that all life on Earth shared a common ancestor, known as the last universal common ancestor (LUCA), more than 3.8 billion years ago (Doolittle 2000; Weiss et al. 2016). The scientific theory of evolution by natural selection, published in Charles Darwin's book *On the Origin of Species* (1859), proposed that evolution is change in the heritable characteristics of biological populations over successive generations, and that “all organisms are derived from common ancestors by a process of branching” (Darwin 1859). Darwin's seminal work was the first to describe this mechanism for evolutionary change, and also championed the theory that evolutionary processes give rise to biodiversity. Biodiversity on Earth has previously been estimated to range widely, from 3 to 100 million species, while more recent numbers suggest that there are closer to a trillion living species, with only 1.9 million species actually named and only 1.6 million species cataloged in databases (Mora et al. 2011; Ruggiero et al. 2015; Loceya and Lennona 2016).

Systematics is the study of the interrelationships of living things. How all these species are named and classified into higher order groups is a branch of science called *taxonomy*. There are many ways to group organisms that have been used in the past and will be discussed below. The focus of this chapter is *phylogenetics* – the field of systematics that focuses on evolutionary relationships between organisms, groups of organisms (e.g. species or populations), or even the genes and proteins found within organisms. A “phylogenetic relationship” between entities such as species, genes, or proteins refers to how those entities shared a common ancestor at some point in the past. Phylogenetic analyses always allow one to infer relationships.

Phylogenetic analysis now commonly uses cladistics – a particular method of hypothesizing relationships among organisms, genes, or proteins. These analyses are based on branching patterns depicted using tree-like representations quite similar to human family trees and are constructed based on similarities in *traits* or *characters*. Traditionally, the characters used in these types of analyses were morphological features of an organism but, with the advent of genomics and the availability of large amounts of whole genome sequencing (WGS) data from a wide variety of organisms, the field has moved to using gene or protein sequences as the basis for these analyses, where each nucleotide or amino acid residue is, quite literally, a “character.”

Genes have been traditionally considered to be heritable units that accumulate mutations over time. Organisms with a shared evolutionary past will have certain mutations in common that can be traced and compared using different algorithms and software. As such, the three core tenets of cladistics can be summarized as follows.

- Any group of organisms, genes, or proteins is related by descent from a common ancestor.
- There is a bifurcating pattern of cladogenesis (clade formation).
- Changes in characteristics occur in lineages over time.

In addition to biological research, an understanding of how genes, proteins, and species are related to each other has many practical applications, such as in bioprospecting, controlling disease outbreaks, forensics, selecting and monitoring drug treatments, tracking ecological degradation, food and agricultural research, and much more. To this end, this chapter will review early classification schemes and the use of molecular sequences as molecular clocks, explain the fundamentals of phylogenetics and interpreting phylogenetic trees (with cautionary notes), describe the differences between common phylogenetic methods and software (and their appropriate use), and provide real-world applications of phylogenetic analyses.

Early Classification Schemes

Throughout history, early classification schemes for naming and organizing taxa have been traditionally based on a range of physiological, morphological, and biochemical characteristics. The Greek philosopher Aristotle (384–322 BC) introduced two key concepts: the classification of organisms by type, and binomial nomenclature (Archibald 2014). Aristotle grouped creatures according to their similarities (e.g. animals with blood and animals without blood), and then organized the groups into a hierarchy. However, Aristotle’s “ladder of nature” (*scala naturae*), or system of classification, was not based on a common evolutionary history, and the various species on the ladder had no specific genetic relationship to each other (Archibald 2014). Aristotle’s binomial definition scheme provided a name for each animal or plant, which consisted of a “genus and difference,” differentiating subgroups of creatures within families according to unique characteristics. However, the use of a formal binomial nomenclature was not applied systematically until two millennia later with the publication of the *Systema Naturae* (1735) by the famous Swedish physician and botanist Carolus Linnaeus (1707–1778), considered to be the father of taxonomy (Linnaeus 1735). By the late twentieth century, Robert Whittaker’s five-kingdom classification system, based mainly upon differences in metabolism, was a standard feature of biology textbooks (Whittaker 1969). The five-kingdom system classified organisms as being members of either the Protista (single-celled eukaryotes), Animalia, Plantae, Fungi, or Monera (unicellular prokaryotes, including most bacteria). Up until this point, classification schemes were largely reliant on morphological or metabolic characteristics or characters that were weighted heavily by an individual scientist, rather than examining the total number of traits organisms had in common.

A more objective approach to biological classification, in which organisms are categorized based on shared derived characteristics due to common ancestry, is provided through the use of cladistics. In cladistics, taxa that share many derived characters are grouped more closely together than those that do not. In this way, these collections of characteristics (or characters)

can be used by scientists to infer phylogenetic or evolutionary relationships. The German entomologist Willi Hennig drafted the seminal work on cladistics, *Basic Outline of a Theory of Phylogenetic Systematics* (1950), while a prisoner of war during World War II (Schmitt 2003). Hennig described how inferred relationships should be shown in a branching hierarchical tree called a cladogram, constructed such that the number of changes from one character state (a branch or clade) to the next is minimized. The tenets of cladistics provide the foundation for modern-day phylogenetic analysis.

Sequences As Molecular Clocks

Nucleic acid sequences in genes and regulatory regions accumulate different types of mutations over time due to a number of mechanisms. These mutations include missense, nonsense, or frameshift errors during DNA replication; insertions and deletions of pieces of DNA, the expansion of repetitive sequences, and even duplication of genes and entire chromosomes (Griffiths et al. 2000). The chemical properties of genetic sequences affect their structure, as well as their ability to interact with other molecules. While mutations in genetic material can impact the function of a cell, downstream consequences of genetic changes also affect the structure, physicochemical properties, and catalytic abilities of proteins (Griffiths et al. 2000). Protein sequence and structure are tightly linked to functionality. As proteins are the workhorses of the cell, changes in their primary sequences can alter cellular and even organismal phenotypes. Some regions of molecular sequences are critical for function. Organisms accumulating mutations in these regions often result in detrimental perturbations in function, thereby reducing fitness. Selection pressure then favors conservation of these regions, while other less critical regions are much more tolerant of change. Rates of change of different positions vary across molecular sequences, between types of genes and proteins, as well as between species and environmental circumstances. In general, the more sequence differences there are between organisms, the more time they likely have had to independently acquire mutations – and, thus, the more distant the evolutionary relationship between them.

The development of protein and nucleic acid sequencing technology in the 1960s and 1970s sparked a profound advance in the ways that scientists could perceive and study organisms. In 1965, Emile Zuckerkandl and Linus Pauling took the opportunity to write an invited (and not peer-reviewed) manuscript to “say something outrageous” about the use of molecular sequences to infer rates of change in evolutionary history, which was later to become known as the molecular clock hypothesis (Zuckerkandl and Pauling 1965). The concept of a molecular clock utilizes the mutation rate of biomolecules to deduce the point in time when two or more life forms, genes, or proteins diverged. Zuckerkandl and Pauling calibrated the rates of amino acid change in human and horse hemoglobin chains using paleontological information to infer the last common ancestor of many animal species. While the study assumed that rates of change of all positions in a sequence are uniform (which is rarely the case in reality), “molecules as documents of evolutionary history” opened the door for implementing DNA and protein sequences for tracing evolutionary events (Zuckerkandl and Pauling 1965).

This concept had a profound impact in microbiology, where the classification of microbes was traditionally based on phenotypic traits that were often subjective. In contrast, sequence comparisons could provide a much more objective, quantitative metric. Indeed, the field of microbiology was further revolutionized in 1977 by Carl Woese, who used 16S ribosomal RNA (rRNA) sequence comparisons to create the modern-day Tree of Life. The Tree of Life classifies organisms within three domains: Eukarya (also called eukaryotes), Bacteria (which he first called eubacteria), and Archaea (initially archaeobacteria) (Woese and Fox 1977; Woese et al. 1990). The 16S rRNA gene (or 18S in Eukarya) seemed an ideal molecular clock, as it was not subject to coding sequence constraints or biases and with certain parts of the sequence mutating at different speeds (Woese and Fox 1977). The faster changing regions could be employed to study more recent relationships, while more slowly changing regions enabled the study of very distant relationships (Woese and Fox 1977).

Woese's idea of using the 16S rRNA gene to construct the Tree of Life has been extended to a variety of other single genes and proteins to construct phylogenetic trees, including rooting the Tree of Life using a duplication within a gene that occurred before the formation of the three Domains of Life (Lawson et al. 1996). Because of different selection pressures, environmental influences, variable accuracy in replication machinery, and other factors, the topologies of these different trees are not always congruent; this reflects the accumulation of changes in gene sequences in different organisms over time. Put otherwise, an organism's evolutionary history is rarely reflected by the history of a single gene's evolution. This observation led to the use of concatenated gene and protein sequence datasets. These are series of different sequences that are linked one after the other, an approach that increases the resolution of phylogenetic signal by gaining consensus among gene histories (Gadagkar et al. 2005). As genome sequencing chemistry and technology continues to improve, WGS has become a powerful tool for understanding biodiversity. The sequences of thousands of genes can now be deciphered and used for phylogenetic analysis and many other applications.

Background Terminology and the Basics

As alluded to above, phylogenetic analysis is the means of inferring or estimating evolutionary relationships. All phylogenetic analyses are based on the analysis of characters or traits. For morphological data, this can be the presence of hair or a certain shape in a bone. Molecular phylogenetics is the study of evolutionary relationships performed by comparing nucleotides or amino acids in a sequence. For sequence data analysis, each column of an alignment is considered to be a character or trait, with each amino acid residue or DNA base in the column representing the particular state of that character.

The resulting relationships are most commonly represented by different types of hierarchical trees. While trees can be depicted in different ways, all contain the basic elements that consist of nodes linked by branches to leaves, connecting ancestors to descendants. A taxon represents a taxonomic group of any rank, such as a species, family, or class, while all of the descendant organisms that originate from a single common ancestor and represent a single monophyletic branch is known as a clade. A cladogram is a tree-based view of evolutionary relationships in which the lengths of the branches in the diagram are arbitrary; in contrast, the branches in a phylogenetic tree often indicate the amount of character change that has occurred. The tree shape, or how the nodes and branches connect the different taxa, is known as the tree topology. These components of a phylogenetic tree are illustrated in Figure 9.1.

The basic steps for constructing a phylogenetic tree include defining a biological question, sourcing and selecting sequences that are homologous (share a common ancestry), a comparison of conserved and variable characters, quantification of the change between sequences, and the representation of the data in a tree. Each of these steps will be discussed in turn below.

Defining the biological question being asked is critical for determining the methods used for analysis and the degree of sampling, defined as the range and types of sequences and species that should be included. It should be noted that not all genes are found in all species. Sequences may be generated in the laboratory or retrieved from private or public databases such as GenBank (National Center for Biotechnology Information (NCBI); see Chapter 1). Whether sequences are generated *de novo* or downloaded from a database, they should be of high quality with few sequence errors and carefully selected to ensure they are homologous. It is also important to acknowledge the source of sequence data when reporting the methods used to generate results. Public databases often employ automated sequence similarity-based algorithms for annotating genomes with gene/protein names. However, different researchers and different organisms often have different naming conventions, so sequences should not be chosen based on their names alone but, rather, based on sequence similarity.

Sequence identity is a quantifiable measure that describes the number of characters that the sequences being compared share that are identical. Sequence similarity is also a quantifiable

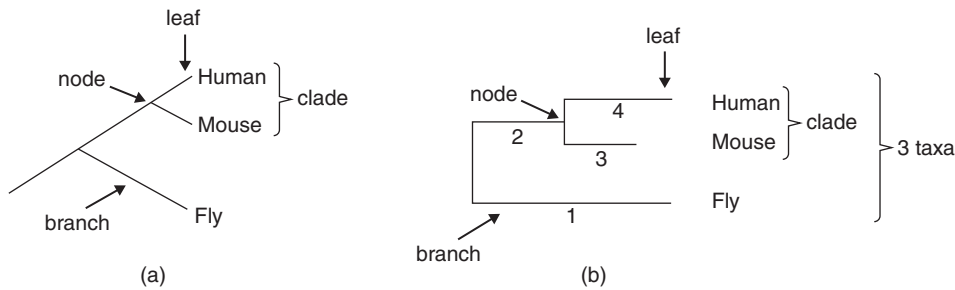


Figure 9.1 Different ways to visualize a tree. In this example, the same tree is presented in both (a) and (b). Taxa are grouped into clades in a tree that comprises a series of branches and nodes that mark bifurcating points in the branches. In (a), note that branch lengths are not significant; they do not indicate the degree of divergence, with the tree only providing the branching order. A clue indicating that a tree is only illustrating the branching order is the equal length of branches and how they align flush with the name of each taxon. (b) The same tree, with branch lengths indicating the degree of divergence that has been inferred from the analysis. By adding up branches between each taxa, one can estimate the degree of divergence between them. In this example, adding up the lengths of branches 1, 2, and 3 indicates the degree of divergence between fly and mouse. Adding up the lengths of branches 1, 2, and 4 indicates the degree of divergence between fly and human. In this artificial example, the differences in branch length would infer that fly and mouse are slightly more related to each other, than fly and human. Note that, in cases such as that shown in (b), only the horizontal branches are significant. The vertical branches are just used to separate out the taxa and make it easier to view them.

measure that describes the number of characters that are either identical or chemically similar in the sequences being compared; keep in mind that this measure does not necessarily reflect ancestry (see Chapter 3). For example, amino acids of the same chemical groups share properties such as charge, polarity, and hydrophobicity. Alanine and valine both have hydrophobic side chains, and so two sequences with these different amino acids at the same position would be considered similar but not identical (Figure 9.2). Sequences that share matching characters (either nucleotides or amino acids) and similar structures because the genes that encoded those sequences were inherited from a common ancestor are called homologs. Homologs share sequence similarity due to inheritance. Homologs in different species that arose from speciation events are called orthologs. However, some species contain multiple copies of a gene (or protein) owing to a process called gene duplication. Once a gene is duplicated within an organism, there is a degree of redundancy in the system that enables selection processes to work differently on the different copies of the gene. The different copies accumulate changes in different ways and at different rates; this results in divergence and often gives rise to new functionality in one or both of the copies. Sequences that are related by gene duplication are called paralogs. Sometimes an organism will acquire a gene from another species through a process called horizontal (or lateral) gene transfer. These different copies of the gene are called xenologs.

It is important to recognize and distinguish these types of relationships when selecting sequences for comparison or during subsequent analysis (Figure 9.3). The best way to select sequences for comparison is through similarity searches performed computationally, such as through a BLAST search (see Chapter 3).

In order to measure the amount of change between different nucleotide or amino acid sequences, they must first be aligned. This is done to ensure the same positions in the gene or protein are being compared. There are different types of alignments that are appropriate for different purposes, and there are many tools for performing both pairwise and multiple sequence alignments (see Chapter 8). Pairwise alignment is a process in which the characters of two sequences are lined up to achieve maximal levels of identity (and conservation, when considering amino acid sequences), allowing one to assess the degree of similarity and the possibility of homology. Multiple sequence alignments are particularly useful for phylogenetic analyses and may focus on a part of a sequence, known as a local alignment, or involve entire sequences, known as a global alignment. Identifying positions that contain many

Alignment 1

Sequence 1 H Q W E R S S T A I I V D C N K D A P R

Sequence 2 H Q W E R S S T V I V V D C T K D L P R

Sequence identity = 16/20 = 80%

Sequence similarity = 20/20 = 100%

Amino acid groups:

A, V, I, L (hydrophobic)
N, T (polar, uncharged)

Alignment 2

Sequence 1 H Q W E R S S T A I I V D C N K D A P R

Sequence 2 R Q W E R S T T A I I V E C N K D A V E

Sequence identity = 15/20 = 75%

Sequence similarity = 18/20 = 90%

H, R (basic)
S, T (polar, uncharged)
D, E (acidic)
P, V (different groups)
R, E (different groups)

Figure 9.2 Alignments illustrating sequence similarity versus sequence identity. Two alignments are shown, comparing sequence 1 with sequences 2 and 3. Alignment 1 compares sequences 1 and 2 and contains four substitutions (highlighted). The substitutions are within the same chemical groups, so they are considered to be similar; A→V, I→V, and A→L are all changes within the hydrophobic amino acids group, and N→T are both polar, uncharged amino acids. There are 16 identical positions and, therefore, 80% identity, while there are 20 similar positions (16 identical plus four similar), for a total of 100% similarity. Alignment 2 compares sequences 1 and 3, and contains five substitutions (highlighted). Three of the substitutions occur within the same chemical group and so they are considered to be similar; H and R are both basic amino acids, S and T are both polar uncharged amino acids, and D and E are both acidic amino acids. However, two substitutions are between chemically unrelated amino acids and so are not considered similar; P and V are from different groups, and R and E are also from different groups. There are 15 identical positions and, therefore, 75% identity, while there are 18 similar positions (15 identical plus three similar), for a total of 90% similarity.

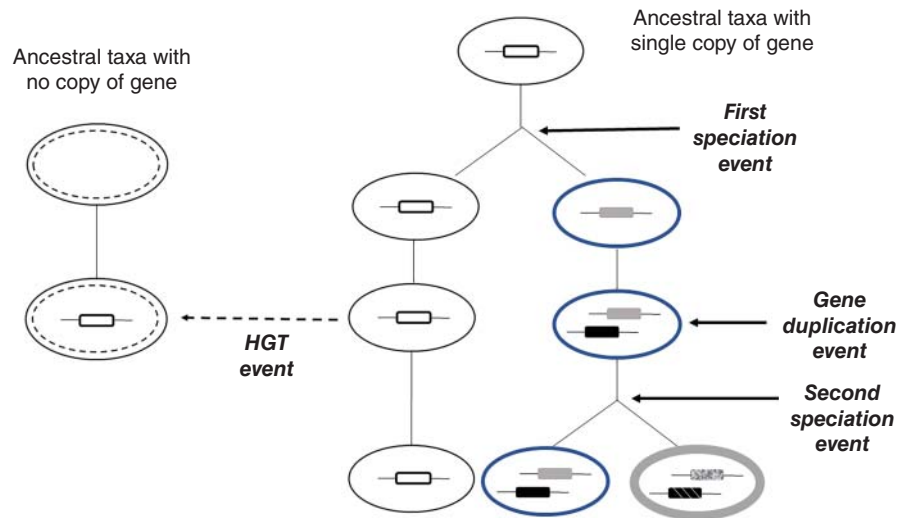


Figure 9.3 The differences between orthologs, paralogs, and xenologs. The ancestral organism on the right contains a single copy of a particular gene (shown as a white rectangle). The descendant lineage (shown in blue, with gray-colored gene) accumulates a number of different mutations resulting in the formation of a new and distinct species in the first speciation event. Each lineage has its own copy of the gene that has differentiated through mutation and selection, and these different versions are called orthologs. A gene duplication event produces two different copies of the gene (duplicated gene is shown in black) in the same organism, which are passed on to descendants and accumulate mutations independently. Genes that diverged only because of gene duplication are called paralogs. The horizontal gene transfer (HGT) event (also called a lateral gene transfer) that delivers a copy of a gene to a new lineage (shown with dashed outline) results in distinct clades or taxa sharing more closely related genes, although the lineages are themselves not as closely related through vertical descent. The sharing of genes through the process of HGT produces xenologs.

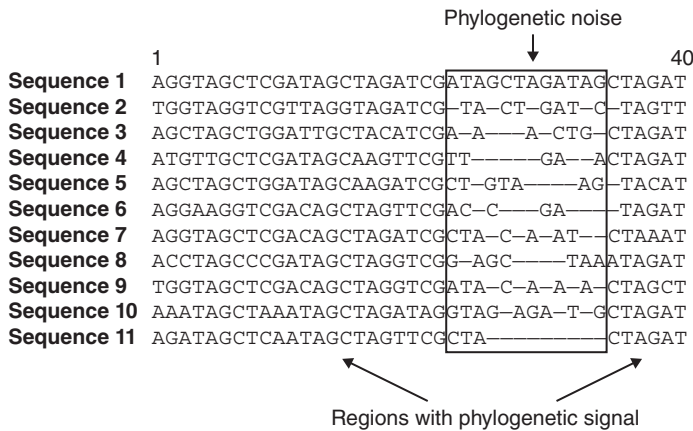


Figure 9.4 The difference between phylogenetic signal and phylogenetic noise. Phylogenetic signal is provided by regions of sequence conservation where different positions can be aligned and contain some variability. These positions contain information about evolutionary processes or rates (here, positions 1–22 and 35–40). Phylogenetic noise is produced by faster evolving sites, which are often difficult to align and may contain several gaps. This noise can mislead phylogenetic inference, resulting in weak support or support for incorrect hypotheses.

identical or highly similar characters, as well as some variable characters, indicates regions of sequence conservation. These conserved regions are more easily aligned and contain the most informative phylogenetic signal. Positions that are highly divergent are often more difficult to align, hence considered to have phylogenetic noise and often not included in most analyses (Figure 9.4). An overview of different types of software, and various considerations for aligning sequences suitable for someone new to phylogenetic analysis, is discussed in the section devoted to tree construction that follows below (see Multiple Sequence Alignment and Alignment Editing).

The path of divergence, or the way sequences have changed over time through the accumulation of mutations, will affect the shape of the phylogenetic tree depicting the inferred course of evolution. The process of quantifying changes between aligned sequences and determining the path of divergence between sequences requires some assumptions, based on the defined biological question one is seeking to answer. Substitution models estimate the likelihood of one base or amino acid changing to another; they also estimate the relative rate of overall change among different sites in the sequence. In general, substitutions are more frequent between bases or amino acid residues that are more similar biochemically. In the case of DNA, the four types of *transitions* (A → G, G → A, C → T, and T → C) are usually more frequent than the eight types of *transversions* (A → C, A → T, C → G, G → T, and the reverse). Such biases will affect the estimated divergence between two sequences.

The specification of relative rates of substitution among particular residues usually takes the form of a square matrix (called a substitution matrix; see Chapter 3). The substitution cost of a more unlikely change is higher than the cost of a more likely change. The off-diagonal elements of the matrix correspond to the relative costs of going from one base to another. The diagonal elements represent the cost of having the same base in different sequences. Different DNA and protein substitution models will be reviewed in the tree construction section below (see Determining the Substitution Model). A factor to note is how genetic changes are propagated from genes to proteins to phenotypic expression. A non-synonymous substitution is a nucleotide mutation that alters the amino acid sequence of a protein. In contrast, nucleotide changes that do not alter amino acid sequences are referred to as synonymous substitutions. As non-synonymous substitutions result in a biological change in the organism, they are more subject to selection.

With the advent of sequencing technologies, a variety of statistical tests have been developed to quantify selection pressures acting on protein-coding regions. Among these, the dN/dS ratio

is one of the most widely used, owing in part to its simplicity and robustness. This measure quantifies selection pressures by comparing the rate of substitutions at silent sites that are presumed neutral (the synonymous substitution rate, or dS) with the rate of substitutions at non-silent sites that possibly experience selection (the non-synonymous substitution rate, or dN). The ratio dN/dS is expected to be greater than 1 only if selection promotes changes in the protein sequence, whereas a ratio less than 1 is expected only if selection suppresses protein changes. Thus, in addition to variation in substitution types, variation in substitution rates among different sites in a sequence has been shown to profoundly affect the results of tree building; this is known as rate heterogeneity (Swofford et al. 1996).

The most obvious example of among-site rate variation, or heterogeneity, is evident among the three positions within a codon in a coding sequence. Owing to the degeneracy of the genetic code, changes in the third codon position are able to occur more frequently without affecting the resulting protein sequence. Therefore, this third codon position tends to be much more variable than the first two. For this reason, many phylogenetic analyses of coding DNA sequences exclude the third codon position. However, in some cases, rate variation patterns are more subtle, particularly those corresponding to conserved regions of proteins or rRNA. Therefore, one should always have as much information as possible about a given gene or sequence before performing any phylogenetic analysis; this includes information on proposed domains, overall degree of conservation, coding and non-coding regions, and RNA structure if analyzing a non-protein-coding gene. To correct for heterogeneity in mutation rates across sites in biomolecules, the gamma distribution can be implemented to model variation (Yang 1994). The gamma distribution is a probability distribution (similar to the better known Poisson distribution) that describes the statistical probability of rates of change, depending on certain parameters. Different forms of the gamma distribution (e.g. the amplitude of the peak and width of the curve) are highly controlled by a single alpha parameter called the “shape parameter.” The higher the value of alpha, the lower the heterogeneity or site variation.

Tree-building methods differ in the details, but essentially all are designed to fit species into related branches and nodes, based on evolutionary models. Tree-building methods can be sorted into distance-based vs. character-based methods. Character-based methods use the aligned sequences directly during tree building. Distance-based methods transform the sequence data into pairwise distances (calculated values which link the most similar sequences together); they then use these derived values rather than the characters directly to build trees (Figure 9.5). While distance-based methods are much less computationally intensive than character-based methods, distance-based methods correct for mutation saturation across sites. Put otherwise, after one sequence of a diverging pair has mutated at a particular site, subsequent mutations in either sequence cannot render the sites any more “different.” In fact, subsequent mutations can make them equal again; for example, if a valine mutates to an isoleucine but then mutates back to a valine, this would result in an “unseen” substitution. These methods also calculate branch lengths, which represent how many changes have occurred between nodes or between a node and a leaf. Long branch lengths

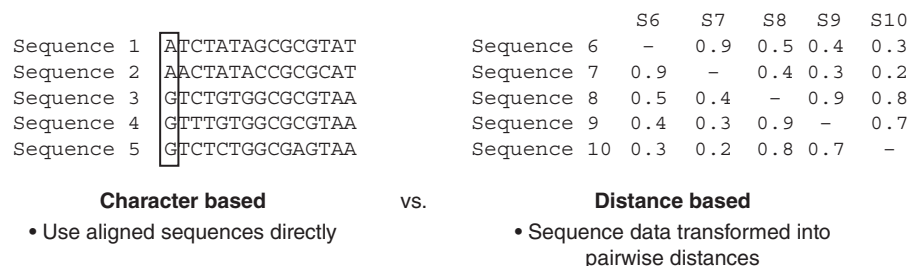


Figure 9.5 Character-based versus distance-based phylogenetic methods. Character-based methods such as Maximum Parsimony and Maximum Likelihood use aligned sequences directly during tree inference, while distance-based methods such as Neighbor-Joining first transform the sequence data into pairwise distances.

indicate more change; shorter branch lengths indicate less change. Different distance-based and character-based methods will be discussed in the tree construction section below (see Tree Building).

There are many different ways to fit data representing species, together in a tree, which generally increases with the numbers of sequences used. Several procedures are available for evaluating the phylogenetic signal in the data and the robustness of the tree topologies. The most popular method involves statistical resampling. It is called bootstrapping. Bootstrapping works on the premise that, if the phylogenetic signal is spread evenly across the sequence, different positions should be sufficiently informative to render the same tree topology (Efron 1979; Felsenstein 1985). Gauging whether this is true or not is important, as some regions of sequence can erroneously influence the tree. For example, domains such as binding cassettes that share sequence similarity but can be found in proteins with very different functionality can adversely influence phylogenetic trees. As such, bootstrapping can be considered a two-step process, where the first step involves generating many newly perturbed datasets from the original set (where the perturbation involves random sampling with replacement) and the second step involves processing the datasets just like in the original phylogenetic analysis. Based on the resulting set of trees (usually 100 or 1000 trees), the proportion of times that a particular branch (e.g. a taxon) appeared in the trees is calculated, and this value is placed on that branch in a consensus tree. This value is commonly referred to as the bootstrap value.

Note that these new datasets are created from the original dataset by randomly sampling columns of characters from the original dataset. This type of random sampling means that each site can be sampled again with the same probability as any of the other sites. As a consequence, each of the newly created datasets has the same number of total positions as the original dataset, but some positions are duplicated or triplicated and others are missing. Therefore, it is possible that some of the newly created datasets are completely identical to the original set – or, on the other extreme, that only one of the sites is replicated, say, 500 times, while the remaining 499 positions in the original dataset are dropped. As a result, bootstrap analysis allows one to identify whether a given branching order is robust to some modifications in the sequence, particularly with respect to the removal and replacement of some sites. Simply put, each bootstrap value acts a measure of confidence in a node.

Phylogenetic trees can be represented as rooted or unrooted. The root of the tree represents the ancestral lineage, and the tips of the branches represent the descendants of that ancestor. As one moves from the root to the tips, one is moving forward in time. Rooting a tree is performed by defining the position on the tree of the (hypothetical) ancestor, usually through the inclusion of an “outgroup,” which can be any organism or sequence not descended from the nearest common ancestor of the organisms or sequences being analyzed. One example of choosing an outgroup might be to use a *Salmonella* sequence as a root for an analysis of a collection of *Escherichia coli* sequences. The *Salmonella* sequence is suitable as an outgroup since it is similar enough to identify phylogenetic signal between all the taxa, but is outside the *Escherichia* genus (the “ingroup”; Figure 9.6).

Similarly, as it is known that reptiles were the progenitors of mammalian species, a reptilian sequence could be used as an outgroup for a mammalian sequence analysis. The outgroup sequence must be selected at the beginning of the phylogenetic analysis and must include in all subsequent steps: alignment, substitution and evolutionary modeling, and tree evaluation. However, outgroup rooting can have issues. An outgroup that is closely related to the ingroup might be simply an erroneously excluded member of the ingroup. A clearly distant outgroup (e.g. a fungus for an analysis of plants) can have a sequence so diverged that its attachment to the ingroup is subject to the “long branch attraction” problem (discussed below, see Tree Building). It is wise to examine resulting tree topologies produced both with and without an outgroup. Another means of rooting involves analysis of a duplicated gene or gene with an internal duplication (Lawson et al. 1996). If all the paralogs from the organisms are included in the analysis, then one can logically root the tree at the node where the paralog gene trees converge, assuming that there are no long branch problems.

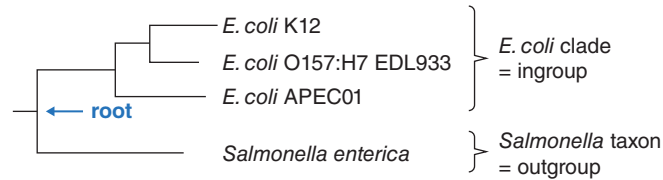


Figure 9.6 Rooting a tree with an outgroup. *Escherichia coli* bacteria are commonly found in the lower intestine of warm-blooded organisms. Most *E. coli* strains are harmless, but some types (called serotypes) are pathogenic and can cause serious food poisoning in humans. Pathogenic lineages (O157:H7 EDL933 and APEC01) of *E. coli* are compared with a wild-type laboratory strain in this small *E. coli* phylogeny. While the relationships between sequences can be inferred from an unrooted tree, the ancestral sequence cannot be inferred unless the tree is rooted. By rooting a tree with an outgroup (in this case a *Salmonella* sequence, which is known to be more distantly related and ancestral to the group under study), it is possible to determine which lineages are ancestral and which are descendants.

Tree viewing software presents nodes, branches, and leaves in different formats, or “views,” and can include rooting, bootstrap, and other confidence metrics, branch lengths, and leaf labeling (e.g. names of taxa, genes, or proteins; sequence IDs, and other information), as required or preferred. In the following section, we provide an overview of tree construction methods and frequently used software for performing phylogenetic analyses.

How to Construct a Tree

Whether it is better to use nucleotides or amino acid datasets for phylogenetic analyses has been the source of some debate, with the debate focusing on the strength of the phylogenetic signal vs. overall ease of use. The main argument for using amino acid sequences to infer phylogeny is that there are more possible character states (20) than nucleotides (four). As such, the increased number of character states can increase resolution during alignment. However, the increased number of characters in nucleotide sequences can lead to better resolution of the tree, particularly when investigating more closely related sequences. Of course, some sequences, such as 16S rRNA sequences, have no associated protein-coding sequence. The decision then falls to the individual performing the analysis, informed by the biological question, the degree of divergence of the sequences being investigated, the available sequences for sampling, and the tools available.

A straightforward phylogenetic analysis consists of four steps: multiple sequence alignment, determining the substitution model, tree building, and tree evaluation. Each step is critical for the analysis and should be handled accordingly. A tree is only as good as the data it is based upon.

Multiple Sequence Alignment and Alignment Editing

Phylogenetic sequence analysis always begins with a multiple sequence alignment. The alignment step in phylogenetic analysis is as important as subsequent steps, if not more important, as it produces the dataset upon which models of evolution are used. Aligned sequence positions subjected to phylogenetic analysis represent a priori phylogenetic conclusions because the sites themselves (not the actual bases) are effectively assumed to be genealogically related or homologous. A typical alignment procedure involves the application of a program such as Clustal (ClustalW, ClustalX, or Clustal Omega), followed by manual alignment editing and submission to a tree-building program (Chenna et al. 2003). Many current methods (including Clustal, PileUp, and ALIGN in ProPack) align sequences according to an explicitly phylogenetic criterion (a “guide tree”) that is generated on the basis of initial pairwise sequence alignments. A widely used algorithm for performing global pairwise alignments is the Needleman–Wunsch algorithm that is implemented in both the Clustal and MUSCLE alignment program packages; this algorithm matches together as many characters as possible between all pairs of sequences

within the input dataset, regardless of their lengths, in order to obtain their pairwise scores (Needleman and Wunsch 1970). Different programs use slightly different approaches to do this. Clustal uses the actual sequences for alignments, while MUSCLE saves time by investigating only k -mers, or short sequences of length k in the sequences, at this first stage of alignment (Needleman and Wunsch 1970). These scores are then used in the construction of a guide tree, which is used to create the multiple sequence alignment.

The aptly named guide tree literally guides the construction of a more robust alignment. The theory is that sequences that are more closely related should be aligned first and then the resulting groups of sequences, which share less relatedness between groups but still have a common ancestor, could then be more accurately aligned with one another. Methods for multiple sequence alignment and examples of commonly used sequence alignment software are discussed in more detail in Chapter 8.

There are many parameters in alignment software that control the speed and sensitivity of comparisons, such as gap penalties and choice of scoring matrix, described more fully in Chapter 3. The most important parameters in an alignment method are those that determine the placement of insert and deletions (indels) or gaps in an alignment of length-variable sequences. Alignment parameters should increase or decrease according to estimated evolutionary divergence, such that base mismatches are more likely as the sequences become more divergent (Thompson et al. 1994). Skewed sampling, such as the over-representation of closely related sequences, can impact pairwise scoring in the guide tree and entrain algorithms, adversely affecting the alignment of under-represented sequences (Thompson et al. 1994; Hughey et al. 1996). Alignment parameters should also be dynamically adjusted in such cases. Dynamic parameter adjustments are available in some software packages, including Clustal. However, unless phylogenetic relationships are known beforehand, there is no clear way to determine which alignment procedure is best for a given phylogenetic analysis.

In general, it is inadvisable to simply subject a computer-generated alignment to a tree-building procedure because the latter is blind to errors in the former. However, as long as the entire alignment is scrutinized in view of independent phylogenetic evidence, methods such as Clustal that utilize some degree of phylogenetic criteria are some of the best currently available. For example, if there are several individual gaps very close to each other in the alignment, they should be grouped into a single indel containing all the gaps since, from an evolutionary standpoint, one insertion or deletion is more plausible than many. Similarly, Clustal encourages the formation of gaps in hydrophilic amino acid sequences, consistent with an insertion or deletion occurring on a globular protein surface, or in the hydrophilic loop regions of a membrane protein, instead of in the hydrophobic protein core. However, it must be emphasized that there are no methods currently available for determining whether one multiple alignment is significantly better than another according to a phylogenetic model.

Alignment of distantly related sequences can be problematic. As discussed earlier, there is an important link between biomolecular structure and function. Often, sequence divergence between distantly related molecules can result in poorly resolved alignments that are either “gappy” or highly variable at many positions. Sometimes, constructing alignments using secondary or tertiary structural information to inform the alignment is considered phylogenetically more reliable than purely sequence-based alignment. This is because confidence in homology assessment is greater when comparing complex characters (such as structures) than simple ones that may have diverged significantly (such as nucleotides and amino acids), resulting in phylogenetic “noise.” This scenario is true in the case of 16S rRNA genes (see Chapter 6). Furthermore, alignment “surgery,” or alignment editing, is sometimes warranted to ensure that phylogenetic signal can be retained and ambiguous information removed; this entails manually removing columns in the dataset. When alignment ambiguities are resolved manually, phylogenetic relationships, substitution processes, and base composition should be considered. It is perfectly reasonable to resolve ambiguities in favor of phylogenetic evidence and, in some cases, to delete ambiguous or noisy regions in the alignment (Figure 9.4). It is useful to perform the phylogenetic analysis based on a series of slightly modified alignments. This is

done to determine how ambiguous regions in the alignment affect the results and what aspects of the results appear to be more reliable.

Determining the Substitution Model

The choice of a substitution model should be given the same emphasis as alignment and tree building. As implied in the preceding section, the substitution model influences both alignment and tree building. Although any of the parameters in a substitution model might prove critical in a given dataset, the best model is not always the one with the most parameters. To the contrary, the fewer the parameters, the better. This is because every parameter estimate has an associated variance or uncertainty. Unfortunately, there is no clear method that is better than another, each one having its own benefits and disadvantages that differ depending on the type of analyses performed and the philosophy of the investigator.

There are a number of different nucleotide substitution models that have been generated by different scientists over the past 50 years. These models estimate nucleotide base frequencies (an estimate of how often a particular nucleotide exists in a sequence) and substitution rates (the rate one nucleotide will be substituted by another as a result of evolutionary processes) differently. The JC69 model (Jukes and Cantor 1969) is the simplest substitution model. JC69 assumes equal base frequencies and equal mutation rates. The only parameter of this model is the overall substitution rate. The K80 model (Kimura 1980) assumes that all of the bases are equally frequent, but distinguishes between transitions and transversions and weights these events differently, thereby affecting the substitution rates. Felsenstein's 1981 model (F81 model) is an extension of the JC69 model, in which base frequencies are allowed to vary (i.e. frequency of $A \neq G \neq C \neq T$; Felsenstein 1981). The HKY85 model by Hasegawa et al. (1985) can be thought of as combining the extensions made in the K80 and F81 models. Specifically, HKY85 distinguishes between the rate of transitions and transversions, also allowing for unequal base frequencies. The T92 model extends Kimura's K80 two-parameter method to the case where a GC-content bias exists (Tamura 1992).

All things being equal, one would expect an organism's GC content to equal 50%, and the consequent AT content to account for the other 50%. However, GC content across species is variable, and the reasons for these differences are thought to be multifactorial and are often controversial. For example, analyses have demonstrated that a correlation exists between GC content and optimal temperature growth for some organisms in certain genomic regions but not for others. Specifically, it has been shown that there is a strong correlation between higher prokaryotic optimal growth temperature and higher GC content of structured RNAs such as rRNA, transfer RNA, and many other non-coding RNAs (Galtier and Lobry 1997; Dutta and Chaudhuri 2010). The TN93 model (Tamura and Nei 1993) distinguishes between the two different types of base transitions (i.e. $A \leftrightarrow G$ is allowed to have a different rate than $C \leftrightarrow T$). Transversions are all assumed to occur at the same rate, but that rate is allowed to be different than both of the rates for transitions. This method is useful when there are strong transition–transversion and GC-content biases, as in the case of the general time-reversible model of Tavaré (1986). This model assumes six substitution rate parameters ($C \leftrightarrow G$, $C \leftrightarrow T$, $C \leftrightarrow A$, $A \leftrightarrow T$, $A \leftrightarrow G$, and $G \leftrightarrow T$) as well as four different base frequency parameters (Tavaré 1986).

In addition to nucleotide substitution models, many amino acid substitution models also exist. The most widely used amino acid replacement models are the PAM (Point Accepted Mutation) and BLOSUM (Block Substitution Matrix) series of matrices (Dayhoff et al. 1978; Henikoff and Henikoff 1992). The details of these replacement models are further discussed in Chapter 3. For phylogenetic analyses, PAM matrices are considered suitable for comparing closely related species, and BLOSUM matrices are generally considered more appropriate for more evolutionarily divergent sequences (Henikoff and Henikoff 1992). For fine-tuned analysis, one may wish to analyze a set of sequences using several scoring matrices to determine the influence of each on the result. Owing to the observed success of the BLOSUM62 matrix

for detecting similarities in distant sequences, this substitution model is used as the default in many sequence search algorithms, such as NCBI's BLAST.

Tree Building

The process of tree building begins with an alignment. Sometimes, the output format of an alignment program is not compatible with tree-building programs and the alignment will require some reformatting; for example, there may be character limits on taxa or molecule name labels and specifications regarding whether sequences must be interleaved or not. As such, data input instructions are important to note before attempting further analysis. Phylogenetic program packages require an alignment, the selection of a substitution model, their accompanying model parameters (with default settings being a good place to start), as well as specifications for bootstrapping and rooting.

As previously discussed, tree-building algorithms are either distance based, which are usually less computationally intensive, or character based. Commonly used distance-based algorithms include Neighbor-Joining (NJ), the Unweighted Pair Group Method with Arithmetic Mean (UPGMA), Fitch–Margoliash (FM), and Minimum Evolution (ME). NJ acts by decomposing an unresolved “star” tree in several iterative steps (Saitou and Nei 1987). The algorithm first identifies the pair of distinct sequences (annotated as taxa, genes, or proteins) with the shortest distance between them, according to the selected substitution model. These taxa, genes, or proteins are then joined to a newly created node that is connected to the central node. The distance from each taxon to the node is calculated, and this value is used to identify the next most closely related sequence, which is then used to create a new node (hence the “joining of neighbors”). This process is repeated iteratively until all of the taxa are resolved into nodes throughout the tree.

Given the variance in substitution models and the differences in bootstrapped datasets, this process can generate different topologies, with differing support at each of the nodes. In this case, a consensus tree, or a tree which contains the most nodes with the most agreement between all possible trees, is identified. Nodes with bootstrap values over 70% are considered well supported by some, while others say only >95% is considered well supported; higher is better. It is important to remember that a high bootstrap value for a node does not mean that the relationship between the taxa (or genes or proteins) is, in fact, true. It simply indicates that that node is supported by the data and the analytical methods selected. Alterations in the alignment, such as the inclusion or exclusion of edited regions of sequence, addition or removal of species, or changes to the computational parameters used can impact the resulting phylogenetic trees to different degrees. Furthermore, the inclusion of a severely misaligned sequence in an alignment may result in very high bootstrap values supporting its separation as a distinct clade, but that is simply due to the misalignment. Manual review of an alignment before phylogenetic analysis is always advised. A well-supported tree inspires confidence in the analysis but, without a time machine to go back and check what actually occurred millions of years ago, an investigator must remember that the result of a phylogenetic analysis simply represents a very good hypothesis. One is always inferring a relationship. This is why the term *phylogenetic inference* is often used.

UPGMA is another clustering algorithm that computes all closest neighbors (Sokal and Michener 1958). This method differs from NJ in that NJ takes average distances to other leaves into account. UPGMA implicitly assumes that all lineages evolve at the same rate (per the molecular clock hypothesis) because it creates a tree where all leaves are equidistant from the root. If the lineages are evolving at different rates (which they do in reality), the UPGMA tree may not fit the distance data very well. As such, UPGMA is generally not considered to be a very good approach for building distance-based trees. The ME and FM methods of phylogenetic inference are based on the assumption that the tree with the smallest sum of branch length estimates is most likely to be the true one (Fitch and Margoliash 1967; Rzhetsky and Nei 1992). FM and ME methods perform best in the group of distance-based methods,

but they work much more slowly than NJ, which generally yields a very similar tree to these methods.

Commonly used character-based algorithms include Maximum Parsimony (MP) and Maximum Likelihood (ML) methods. The parsimony principle, basic to all science, posits that, all things being equal, the simplest possible explanation is the best. In terms of tree building, the MP method requires the fewest evolutionary changes or the fewest number of character-state changes (Swofford et al. 1996). All of the character data are used in the analysis; however, no branch lengths are calculated while the relationships between sequences are determined. Although it is easy to score a phylogenetic tree by counting the number of character-state changes, there is no algorithm to quickly generate the most parsimonious tree. Instead, the most parsimonious tree must be found in what is commonly referred to as “tree space,” meaning among all possible trees. MP analyses tend to yield numerous trees, often in the thousands, which have the same score but different topologies. In this case, the tree with the topology containing the most nodes in consensus with all equally likely trees is considered to be the one that best supports the data.

When a small number of taxa are considered for MP, it is possible to do an exhaustive search in which every possible tree is scored and the best one is then selected. For greater numbers of taxa, a heuristic search that involves finding an approximate solution when an exact solution is not feasible must be performed. It should be noted that the MP method performs poorly when there is substantial among-site rate heterogeneity (Huelsenbeck 1995). Also, an optimal MP tree will minimize the amount of homoplasy – convergent evolution where characters have evolved independently. As such, MP methods sometimes suffer from long branch attraction (Bergsten 2005). Long branch attraction can occur when different rapidly evolving lineages are misinterpreted to be closely related, regardless of their true relationships. Often, this situation arises because convergent evolution of one or more characters included in the analysis has occurred in multiple taxa. MP programs may erroneously interpret this homoplasy as a synapomorphy, evolving once in the common ancestor of the two lineages.

In contrast, ML methods seek to find the tree that best explains the data given a particular model of sequence evolution, specified by parameter and distribution settings by the user. Quartet puzzling is a relatively rapid tree-searching algorithm available for ML tree building (Strimmer and von Haeseler 1996). With ML, the simplest explanation may not be considered the most correct if additional information is known about the dataset (e.g. high rates of change across sites). While the ML method is very slow and computationally demanding, it is thought to produce the best representations of evolutionary processes. As such, the ML approach has been the basis of a powerful statistical method known as Bayesian inference (Huelsenbeck et al. 2002).

Bayesian inference is a method of statistical inference in which the probability for an evolutionary hypothesis is updated as the algorithm progresses and more evidence or information becomes available. The updated probability of an outcome is determined from a prior probability and a likelihood function. A prior probability is a set of parameters and distributions for an outcome, which are determined before any data are examined. The prior probability helps determine the chances of possible outcomes prior to knowing anything about what actually happened. The “likelihood function” consists of sets of parameters and distributions for an outcome when things are known about what could have happened. During analysis, updates in the probability of an outcome occur through the use of a Markov chain Monte Carlo algorithm that iteratively compares samples of likelihoods of outcomes (and their sets of parameters and distributions) with the data and parses out the most likely outcomes; this then informs the range of likelihoods to be further sampled (Yang and Rannala 1997). This process occurs as many times as the investigator prescribes. Bayesian methods use the same approach as ML in that the tree that best represents the data according to a model of evolution is considered the “best tree”; however, the likelihood calculation is considered to be “more informed.”

Table 9.1 Some common software packages implementing different phylogenetic analysis methods.

Software package	Description
BEAST	<ul style="list-style-type: none"> • Cross-platform program for Bayesian analysis of molecular sequences using Markov chain Monte Carlo • Produces rooted, time-measured phylogenies inferred using strict or relaxed molecular clock models
MEGA	<ul style="list-style-type: none"> • User-friendly Windows-based platform for sequence upload, alignment (ClustalW or MUSCLE), and phylogenetic inference by a variety of methods (maximum likelihood, evolutionary distance, and maximum parsimony)
MrBAYES	<ul style="list-style-type: none"> • Program performing Bayesian inference of phylogeny using a variant of Markov chain Monte Carlo
PHYLIP	<ul style="list-style-type: none"> • Menu-based package of 35 different programs for inferring evolutionary trees • Parsimony, distance matrix, and likelihood methods, bootstrapping and consensus trees • Data types that can be handled include molecular sequences, gene frequencies, restriction sites and fragments, distance matrices
PhyML	<ul style="list-style-type: none"> • Fast program for searching Maximum Likelihood trees • Uses nucleotide or amino acid sequences
PAUP	<ul style="list-style-type: none"> • Phylogenetic Analysis Using Parsimony (and other methods later than v4.0) • Available as a plugin for Geneious

Popular software implementing these different types of methods are described in Table 9.1. An example workflow of an NJ DNA sequence analysis using the classic PHYLIP program package is shown in Figure 9.7.

When building a phylogenetic tree, it is important to look at the data from as many angles as possible. Consistency of tree topologies generated by different methods suggests that the analysis is a good estimate for the true phylogeny. Unfortunately, consistency among results obtained by different methods does not necessarily mean that the result is statistically significant or represents the true phylogeny, as there can be several reasons for such correspondence. The choice of outgroup taxa can have as much influence on the analysis as the choice of ingroup taxa. In particular, complications will occur when the outgroup shares an unusual property (such as composition bias or clock rate) with one or several ingroup taxa. Therefore, it is advisable to compute every analysis with several outgroups and check for congruency of the ingroup topologies. Also, be aware that programs can give different trees depending on the order in which the sequences appear in the input file. PHYLIP, PAUP, and other phylogenetic software provide a “jumble” option that reruns the analysis with different (jumbled) input orders.

If, for whatever reason, a tree must be computed in a single run, sequences that are suspected of being “problematic” should be placed toward the end of the input file to lower the probability that tree rearrangement methods will be negatively influenced by a poor initial topology stemming from any problematic sequences. In general, one should always consider any bioinformatic analysis in an evolutionary context when it is based on evolutionary assumptions. For example, if a BLAST analysis was performed, one should ask questions such as: Which of the hits in the BLAST analysis are likely orthologs versus paralogs? Which of the membrane proteins identified in a search are likely homologs (ancestrally related) versus similar by chance due to similarities in trans-membrane alpha-helical sequences? What domains seem to be conserved in a set of aligned sequences? Are there indels associated with one clade and not another, indicating that they may have functional significance?

Tree Visualization

There are several parts that make up the anatomy of a phylogenetic tree. The skeleton of the tree consists of the nodes, branches, leaves, and (if included) the root. Labeling of leaves usually corresponds to gene, protein, or species names, but can also include common names of organisms,

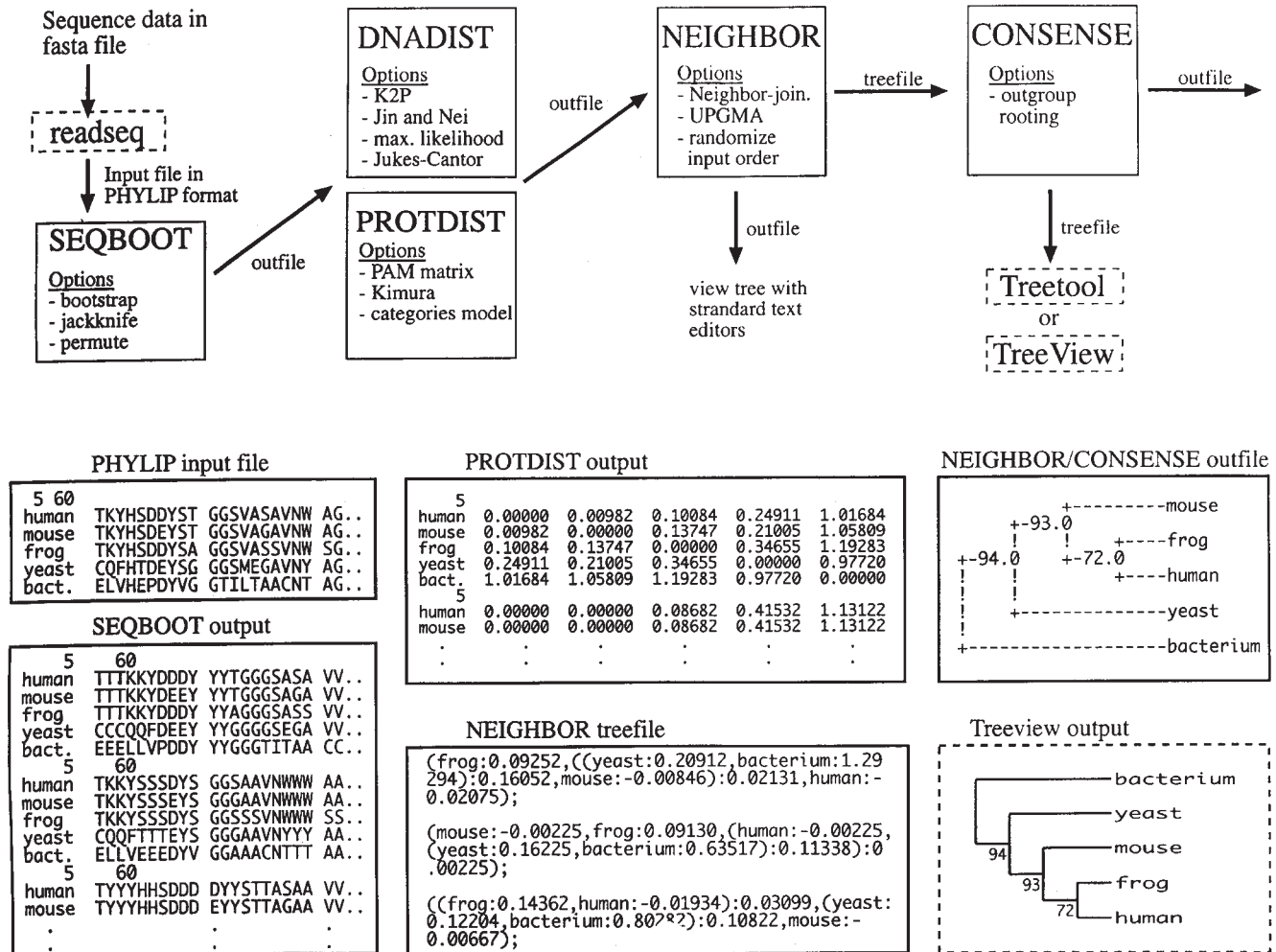


Figure 9.7 Workflow for a protein-based phylogenetic analysis using the PHYLIP program package. Protein sequences in FASTA format are converted into PHYLIP format by READSEQ (which is not part of PHYLIP but is freely available online). SEQBOOT accepts the PHYLIP file as input and sequences are bootstrapped a user-defined number of times. (For the purposes of this example, assume that the user has specified 1000 bootstraps.) The resulting outfile can be used to calculate 1000 distance matrices for input into PROTDIST. In this step, the actual amino acids are discarded and replaced by a calculated value that is a measure of the amount of divergence between the sequences. The NEIGHBOR program joins nodes and branches according to these calculated values, creating 1000 trees from these matrices. The CONSENSE program reduces the 1000 trees to the one that includes only those nodes that are present in the majority of the trees in the set of all possible trees and indicates the bootstrap values by the nodes. TREEVIEW or TreeTool allow the user to manipulate the tree (e.g. rerooting, making branch rearrangements, and changing fonts) and to save the file in a number of commonly used graphic formats. Although TREEVIEW and TreeTool are not part of the PHYLIP program package (indicated by boxes with dashed lines), they are freely available. The figure also shows the different file formats used during processing through the various stages of bootstrap analysis. Periods to the right and at the bottom of a box indicate that files were truncated to save space.

sequence accession numbers, or types of characteristics under investigation. Species names can be formatted using bolded, italicized, or color-coded characters. A typical phylogeny will also include bootstrap values positioned beside their respective nodes, as well as a branch length scale bar. This is a bar at the bottom of the figure accompanied by a number, usually a fraction, that calibrates the number of changes per given number of characters. Branch lengths can also be quantified and labeled on the tree. Branches, nodes, and leaves cannot generally be removed from a tree visualization without removing those sequences from the alignment and performing the analysis anew.

Phylogenetic trees can be visualized in different ways. For example, trees can be drawn horizontally, vertically, circularly, or radially (Figure 9.8). Leaves and branches can rotate

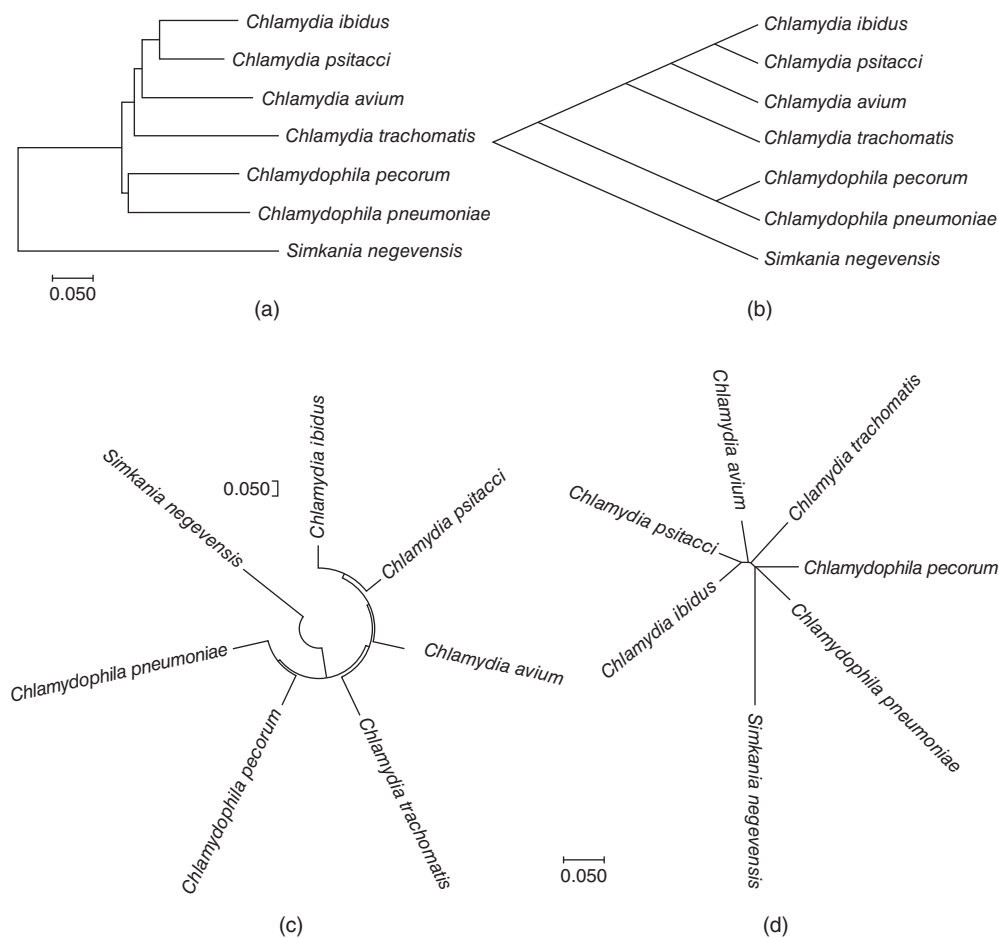


Figure 9.8 Phylogenetic relationships can be visualized using different types of tree views; however, the relationships between species are the same. The Chlamydiales order of bacteria contain both human (*Chlamydia trachomatis*, *Chlamydophila pneumoniae*) and animal (*Chlamydia psitacci*, *Chlamydophila pecorum*) pathogens, as well as lineages that are pathogenic to both (*Simkania negevensis*). The phylogeny of this subset of Chlamydiales species is presented as a phylogram (a), as well as in linear (b), circular (c), and radial views (d). For example, in all representations, *Chlamydia psitacci* is always shown to be most closely related to *Chlamydia ibidus*, while *Simkania negevensis* is always shown to be the most distantly related species.

about nodes without altering the relationships inferred. There are a number of tree-drawing programs currently available for use on a variety of computing platforms, including TreeTool, TreeDraw, PHYLODENDRON, TREEVIEW, FigTree, and the tree-drawing tool within PAUP; all of these handle standard tree files. These programs facilitate not only the generation of trees suitable for publication or other presentation but also facilitate viewing of the data in general. For example, programs such as the freely available TREEVIEW enable the user to manipulate the branching order view, root the tree, and perform other graphical manipulations that can aid the user.

A more extensive list of phylogenetic tree-viewing software, including web-based views of precomputed trees, can be found by following the Phylogenetic Tree Visualization Software link in the Internet Resources section. Tree images/files can also often be exported to other commonly used presentation and graphics software such as PowerPoint or Photoshop and overlaid with other characteristics of biological relevance, such as clusters of phylogenetically related isolates involved in a disease outbreak investigation that are distinguished from sporadic cases of illness.

Marker-Based Evolution Studies

Genetic molecular markers are fragments of DNA that are associated with certain locations within the genome. Molecular markers have been used to diagnose diseases such as cystic fibrosis, resolve taxonomic affinity using 16S rRNA genes, and are used in molecular biology as DNA barcodes, enabling the identification of particular sequences in a pool of unknown DNA. There are many different types of molecular markers that can be used to generate hierarchies of relationships between organisms or characteristics, such as the predisposition for disease. As discussed, these markers can consist of single-nucleotide variants (SNVs) in genes or amino acid substitutions in different proteins. While indels can be arbitrary inserts or deletions, conserved signature indels are defined as only those protein indels that are present within conserved regions of proteins; they are also restricted to a particular clade or group of species (Gupta and Griffiths 2002). Conserved signature indels provide useful molecular markers for inferring evolutionary relationships, as it is unlikely that the same insertion or deletion event occurred at the same position in two independent evolutionary lineages.

SNVs can also be traced and compared between entire genomes to group sequences together in a number of different ways. SNV markers include sequence motifs or short recurring patterns in DNA that are presumed to have a biological function, such as those found in transcription factor binding sites. When SNVs occur within enzyme restriction sites, they can affect genomic DNA digestion patterns that can be detected using a technique called pulsed-field gel electrophoresis (PFGE; Gerner-Smidt et al. 2006). Different types of isolates with identical digested fragment patterns are considered the most closely related. The PFGE method has been used for molecular microbial typing; it has also been used for the classification of isolates (at the subspecies level) from clinical or environmental samples, such as foodborne pathogens for outbreak investigation. Another microbial typing technique, known as multi-locus sequence typing (MLST), classifies different patterns of SNVs at particular genetic loci to assign microbial isolates to “sequence types” based on the sequencing of DNA fragments rather than their electrophoretic mobility as in PFGE. MLST can be performed on a standard set of housekeeping genes and is used to characterize strains by their unique allelic profiles (Margos et al. 2008). Alternatively, MLST can be performed on whole or core genomes, and the vast number of alleles produced by this method are then compared using a matrix of pairwise differences that are displayed as a tree (Achtman et al. 2012). MLST relationships are often visualized using what are called minimum spanning trees; these trees connect all the nodes by the shortest possible path. Minimum spanning trees cluster sequence types together and attempt to identify the founding (or ancestral) sequence type of each group (Salipante and Hall 2011). The ancestral types are then connected in a radial view (Figure 9.9).

The molecular typing of different microbial organisms is based on different MLST schema (collections of loci and alleles), as different loci are more informative than others in different lineages owing to different rates of change and selective forces. Some schema have the power to represent biological phenomena *in silico*, such as serotypes – the immunological properties at the cell surface that can be used to distinguish different strains. A web-accessible tool for *Salmonella* serotype prediction based on the core genome MLST schema is called SISTR (for *Salmonella* In Silico Typing Resource; Yoshida et al. 2016). Such tools enable rapid identification of *Salmonella* contamination to support food safety and public health investigations. With the increasing adoption of genomic analyses in epidemiology to understand the distribution and spread of infections, software such as eBURST (Feil et al. 2004) and SISTR, as well as the databases housing schema and isolate data, will be critical for reducing the number of preventable cases of infectious disease.

Although most eukaryotic DNA is packaged in chromosomes within the nucleus, mitochondria also have a small amount of their own DNA. Mitochondrial DNA (mtDNA) is a small double-stranded DNA found in most eukaryotes (e.g. human mtDNA contains only 37 genes) and is maternally inherited (Anderson et al. 1981). As animal mtDNA evolves faster than nuclear genes (Brown et al. 1979), it carries SNVs which are valuable tools in the fields of forensic, population, and medical genetics (Kundu and Ghosh 2015; Sturk-Andreaggi et al.



Figure 9.9 Excerpt of a *Salmonella* minimum spanning tree. Types of *Salmonella* bacteria cause food poisoning associated with diarrhea, fever, abdominal cramps, and vomiting. *Salmonella* sequence types (STs) from an outbreak are shown clustered together in star shapes to attempt to identify the founding ST of each group of infections. The radiations off the circular hubs represent the closest relatives of the founders. The size of the circle is proportional to the number of sequences with the same ST, while the color of the circle represents different sources of bacteria (e.g. food product, environment, or sample type). Epidemiologists and researchers can use this information to identify the source of *Salmonella* contamination, and prevent further infections. Image courtesy of Nabil Fahreed-Alikhan (created using EnteroBase software, Warwick University, UK).

2017; Theurey and Pizzo 2018). An example of a tools for examining relationships between mtDNA sequences is mtDNAprofiler (Yang et al. 2013).

Plant-based molecular marker studies have been used for crop improvement. A number of functional molecular markers have been developed which are readily identified by genetic sequence analyses in wheat, rice, maize, sorghum, millets, and other crops (Kage et al. 2016). For example, alleles identified in 30 different genes in wheat have been associated with food quality, agronomic, and disease resistance traits, and used successfully in breeding programs (Liu et al. 2012). Plant-based molecular marker studies famously led to advancements in agricultural productivity in the world's food supply in the 1960s, known as the Green Revolution (Hedden 2003). Point mutations in wheat *Rht1* and *Rht2* genes enabled "dwarfing" of plants, which increased stalk strength and consequently grain yield (Hedden 2003). Assays generated based on such analyses continue to enable farmers and scientists to screen new cultivar genotypes for desired characteristics.

Phylogenetic Analysis and Data Integration

Phylogenies and evolutionary analyses are used to answer many types of biological questions. For example, the function of hypothetical proteins can be inferred from branch patterns when the protein under study clusters closely with well-annotated sequences in a process called function prediction. Similarly, differences in branch patterns originating from sequence divergence between orthologous and paralogous proteins may indicate a divergence of

function. Different phylogeny-based protein function prediction tools are available, such as SIFTER (Statistical Inference of Function Through Evolutionary Relationships; Sahraeian et al. 2015), although many function prediction algorithms are built on alignment-based similarity; these include BLAST and PredictProtein. In many cases, different types of data from multiple sources must be integrated with phylogenetic information to answer a biological question. The frequency of SNVs can be assessed in populations, and their spread across geographical regions can be understood through a discipline known as phylogeography. GenGIS is a software platform that merges geographic, ecological, and phylogenetic biodiversity data together in order to visualize phylogenetic relationships across a variety of environments. GenGIS has been used to assess taxonomic diversity from the Global Ocean Sampling expedition (Parks et al. 2009) and the spread of HIV-1 subtypes across Africa (Parks et al. 2013). Similarly, MicroReact integrates genomics data with temporal, geographic, and other metadata to create health-related visualizations (Argimón et al. 2016). This platform has been used to reconstruct the Western Africa Ebola epidemic and transmission events of various multi-drug-resistant organisms around the world. In addition to research tools, personal genomics companies such as 23andMe (see Internet Resources) use marker genes and phylogeographic analyses to identify health risks and trace family ancestry around the world.

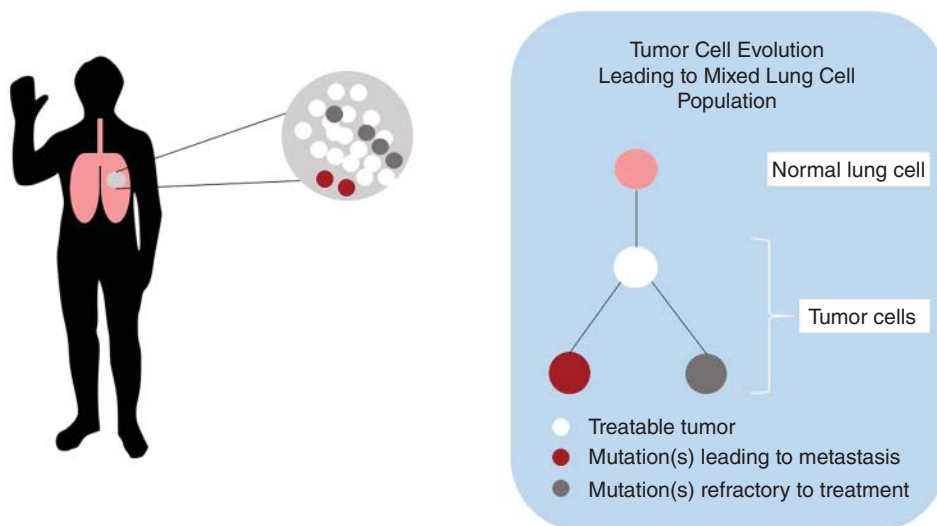
Genomic epidemiology applies WGS data to understand how genomic variation within microbial populations (both microorganisms and viruses) affects the incidence, distribution, and possible control of diseases and other factors relating to public health. The genomes of microbial isolates, as well as clinical, exposure, geographic, and demographic data, are compared in phylogenomic trees and other comparative tools (Tang et al. 2017). Patients infected with isolates in clusters believed to be involved in outbreaks are investigated for common sources of infection and modes of transmission, and this information is then used to control the spread of disease (Robinson et al. 2013). Genomic epidemiology techniques have been used worldwide for control of many types of infectious diseases such as tuberculosis, *Salmonella*, *E. coli*, and various viral diseases (Gardy et al. 2011; Croxen et al. 2017; Moran-Gilad et al. 2017). For example, scientists involved with the 2016 Singapore Zika outbreak used genomic epidemiology techniques to match hospital and mosquito viral strains through Bayesian analysis; these results were then used to guide subsequent prevention measures, such as where to increase larvicide deployment and where public awareness programs should be initiated (Singapore Zika Study Group 2017).

Phylogenetics has also proven useful in the emerging field of microbial forensics, which serves to link microbial DNA evidence from acts of bioterrorism or inadvertent microorganism/toxin release to potential sources for attribution purposes (Schmedes et al. 2016). For example, in 2001, the U.S. Postal Service was the target of an anthrax bioterrorism attack. Precise strain genotyping and phylogenetic analysis clustered sequences from seemingly disparate infections in Connecticut, New York, Florida, and Washington, DC, to a single perpetrator while eliminating cases due to natural causes (Yang and Keim 2012). As U.S. health and law enforcement officials pursued the perpetrator, knowing the exact strain type of *Bacillus anthracis* was invaluable for narrowing the potential sources and for defining the crime scene itself (Yang and Keim 2012). Advancements in sequencing technologies and bioinformatic analyses continue to influence policies and practices with regards to biodefense, criminal investigations, and intelligence acquisition (Schmedes et al. 2016).

Cancer is a genetic disease that arises when normal cellular functions are disrupted by mutations arising in DNA. Cancer research involves a range of clinical and epidemiological data, as well as molecular and evolutionary analytical approaches. Mutations occur at the level of single cells and are then propagated into subpopulations as cells divide. Differences in growth rates in subpopulations produce a complex tumor microenvironment consisting of many different interacting and evolving cells (Beerenwinkel et al. 2016). The resultant intratumor genetic diversity poses a huge problem for correctly diagnosing and treating tumors, especially as a biopsied sample may not be representative of the entire tumor (Beerenwinkel et al. 2016). Tumor phylogenetics provides insights into evolutionary mechanisms causing disease, and is also providing insight into the prediction and control of cancer progression, metastasis, and therapeutic responses (Box 9.1).

Box 9.1 Predicting Cancer Progression and Drug Response Using Phylogenetic Approaches

Tumor phylogenetics provides insight into evolutionary mechanisms causing disease. Cancer is a genetic disease in which the accumulation, diversification, and selection for mutations are now known to promote tumor cell proliferation and impact survival according to complex evolutionary mechanisms. The figure shows how tumors can contain a mixed population of cells that have accumulated different types of mutations. Some mutations enable cancer cells to metastasize, while others render cancer cells less susceptible to treatment. Phylogenetic analysis has been applied to the understanding of predicting and controlling cancer progression, metastasis, and therapeutic responses. Tumor phylogenetics aims to reconstruct tumor evolution from genomic variations by exploring the space of possible trees in order to explain a dataset. In particular, evolutionary theory and analyses have been developed for determining the heterogeneity of tumor cells, specifically the types of mutations associated with different clinical outcomes; these include copy number variants, microsatellites (tracts of repetitive DNA in which certain DNA motifs are repeated), and “mutation signatures” such as nucleotide biases linked to environmental triggers. Rates of mutation, as well as the extent and intensity of selective pressures, greatly affect treatment options and prognosis. Different treatment regimens lead to selection that can, in turn, alter the dominant clones within tumors. Single-agent treatment can lead to relapse by selecting for non-responsive clones and higher mutation rates (intratumor heterogeneity), and has been linked with the ability to resist different types of therapy. These types of tumor diversity studies depend highly on appropriate parameter estimation and modeling of different mutational processes that have been validated with observed data. Most studies of tumor phylogenetics to date have adapted standard algorithms that were developed for generating phylogenies of different species (Schwartz and Schaffer 2017).



Phylogenetics is also being used to advance pharmaceutical development through the newly emerging field of pharmacophylogenomics (Searls 2003). Pharmacophylogenomics is a field of study that combines knowledge about genes, protein localization, gene/protein relatedness, drugs, and drug targets to identify novel sources of therapeutics (Searls 2003). One of the best known pharmacophylogenomics discoveries is the “druggable genome” – the identification of the ~3000 genes in the human genome that express proteins able to bind drug-like molecules (Hopkins and Groom 2002; Sakharkar et al. 2007). For proteins that are both interacting and evolving, such as receptors and peptide ligands (i.e. chemokines and their G-protein-coupled

receptors), co-evolution is reflected in similarities in the topologies of their phylogenetic trees (Searls 2003). Studies identifying evolutionary trends can be used to create algorithms for the de novo prediction of molecular interactions. As pathways and networks often co-evolve in parallel with their interacting partners, these studies also expand phylogenetic analysis from genes and proteins to entire metabolic and physiological networks, widening the search for potential sources of new drugs (Searls 2003).

The fields of metagenomics and metabolomics (see Chapter 14) are also rapidly expanding our ability to explore the genetic diversity of novel terrestrial and aquatic environments. Metagenomics itself is the study of genetic material recovered directly from an environmental sample, explores the diversity of complex microbial ecosystems, including strains which cannot be cultured (Handelsman 2004). The NCBI public repository offers access to sequences from a wide range of environmental communities, including submerged whale carcasses, sludge, farm soil, acid mine drainage sites, subtropical gyres, and deep-sea sediments, to name a few (NCBI Resource Coordinators 2016). Phylogenetic profiling of these gene repertoires provides an *in silico* form of analysis, which can help to focus the direction of *in vitro* experimentation. For example, the characterization of the diversity and prevalence of bacterial resistance gene products targeting β -lactams and A- and B-type streptogramins were initially identified in human pathogens (D'Costa et al. 2007). Through metagenomic phylogenetic analysis, sequences were also found in many environmental species, suggesting an underappreciation of the soil resistome, which was also supported by *in vitro* studies (D'Costa et al. 2007). Such findings provide much motivation for better antibiotic stewardship and the judicious use of antibiotics in the clinic.

Future Challenges

Phylogenetic analysis is a powerful tool for answering many types of biological questions. Phylogenetic trees, however, are inferred, dynamic constructs – they depend on the methods used, the regions of sequences included/excluded, the sampling of species, the parameters, the rooting, and other factors. Paradoxical as it may sound, by far the most important factor in inferring phylogenies is not the method of phylogenetic inference but the quality of the original data. The importance of data selection and of the alignment process cannot be overestimated. Even the most sophisticated phylogenetic inference methods are not able to correct for biased or erroneous input data. As such, an investigator should always look at the data and the results of any analyses from as many angles as possible, checking that the results make general biological sense.

As DNA sequencing technology continues to decrease in cost and improve in speed, read length, and accuracy, so must the capacity to curate, analyze, store, and share sequence data. Tools and integrative platforms for performing phylogenetic and other bioinformatic analyses continue to proliferate as scientists innovate new uses and applications of sequence information. In the era of “big data,” the barriers for phylogenetics and bioinformatics lie not in the ability to produce data, but rather in the availability of individuals with sufficient expertise to perform analyses, as well as in the infrastructure needed to perform the computations (Muir et al. 2016). As such, analysts and bioinformaticians with the skills to carry out phylogenetic analyses of genes, genomes, proteins, and other types of molecular and systems information are, and will continue to be, in demand. Furthermore, the accuracy, sensitivity, and specificity (see Box 5.4) of tools and algorithms must be systematically and quantitatively assessed in order to characterize the different strengths and weaknesses of each. This will allow the community to decide which tools and algorithms are the most appropriate and how the results from each can be compared and integrated.

Going forward, different types of integrative bioinformatic and phylogenetic analyses of the vast amount of available data will provide new ways to understand our world, and teach us new ways to adapt to our ever-changing environment. The evolution of phylogenetics, as well

as life on Earth, is well summarized by one of the most popular misquotes of Charles Darwin that is placed in the stone floor of the headquarters of the California Academy of Sciences: “It is not the strongest of the species that survive, nor the most intelligent, but the one most responsive to change.”

Internet Resources

ALIGN	www.sequentix.de/software_align.php
BEAST	beast.community
BLAST (NCBI)	blast.ncbi.nlm.nih.gov/Blast.cgi
ClustalW/ClustalX	www.clustal.org/clustal2
eBURST	eburst.mlst.net
Enterobase	enterobase.warwick.ac.uk
FigTree	tree.bio.ed.ac.uk/software/figtree
GenGIS	kiwi.cs.dal.ca/GenGIS/Main_Page
MEGA	www.megasoftware.net
Microreact	microreact.org/showcase
MrBayes	mrbayes.sourceforge.net
MUSCLE	www.drive5.com/muscle
mtDNAprofler	mtprofler.yonsei.ac.kr
PAUP	paup.phylosolutions.com
PHYLIP	evolution.genetics.washington.edu/phylip.html
Phylogenetic Tree Visualization Software	en.wikipedia.org/wiki/List_of_phylogenetic_tree_visualization_software
PhyML	www.atgc-montpellier.fr/phyml
PileUp	www.biology.wustl.edu/gcg/pileup.html
PredictProtein	www.predictprotein.org
SIFTER	sifter.berkeley.edu
SISTR	lfz.corefacility.ca/sistr-app
TreeDraw	webconnectron.appspot.com/Treedraw.html
TreeTool	github.com/neherlab/treetool
TREEVIEW	taxonomy.zoology.gla.ac.uk/rod/treeview.html
23andMe	www.23andme.com

References

- Achtman, M., Wain, J., Weill, F.X. et al., and the S. Enterica MLST Study Group (2012). Multilocus sequence typing as a replacement for serotyping in *Salmonella enterica*. *PLoS Pathog.* 8: e1002776.
- Allegre, C.J. and Schneider, S.H. (2005). Evolution of Earth [online]. *Sci. Amer.* 293.
- Anderson, S., Bankier, A.T., Barrell, B.G. et al. (1981). Sequence and organization of the human mitochondrial genome. *Nature.* 290 (5806): 457–465.
- Archibald, J.A. (2014). *Aristotle's Ladder, Darwin's Tree: The Evolution of Visual Metaphors for Biological Order*. New York, NY: Columbia University Press.
- Argimón, S., Abudahab, K., Goater, R.J.G. et al. (2016). Microreact: visualizing and sharing data for genomic epidemiology and phylogeography. *Microb. Genom* 2 <https://doi.org/10.1099/mgen.0.000093>.
- Berenwinkel, N., Greenman, C.D., and Lagergren, J. (2016). Computational cancer biology: an evolutionary perspective. *PLoS Comput. Biol.* 12 (2): e1004717.

- Bergsten, J. (2005). A review of long-branch attraction. *Cladistics*. 21: 163–193.
- Bouvier, A. and Wadhwa, M. (2010). The age of the solar system redefined by the oldest Pb-Pb age of a meteoritic inclusion. *Nat. Geosci.* 3: 637–641.
- Brown, W.M., George, M. Jr., and Wilson, A.C. (1979). Rapid evolution of animal mitochondrial DNA. *Proc. Natl. Acad. Sci. USA*. 76: 1967–1971.
- Chenna, R., Sugawara, H., Koike, T. et al. (2003). Multiple sequence alignment with the Clustal series of programs. *Nucleic Acids Res.* 31: 3497–3500.
- Croxen, M.A., Macdonald, K.A., Walker, M. et al. (2017). Multi-provincial Salmonellosis outbreak related to newly hatched chicks and poults: a genomics perspective. *PLoS Curr.* 9: 9.
- D’Costa, V.M., Griffiths, E., and Wright, G.D. (2007). Expanding the soil antibiotic resistome: exploring environmental diversity. *Curr. Opin. Microbiol.* 10: 481–489.
- Darwin, C. (1859). *On the Origin of Species*. London, UK: John Murray.
- Dayhoff, M.O., Schwartz, R.M., and Orcutt, B.C. (1978). A model of evolutionary change in proteins. In: *Atlas of Protein Sequence and Structure* (ed. M.O. Dayhoff), 345–362. Washington, DC: National Biomedical Research Foundation.
- Dodd, M.S., Papineau, D., Grenne, T. et al. (2017). Evidence for early life in Earth’s oldest hydrothermal vent precipitates. *Nature*. 543: 60–64.
- Doolittle, W.F. (2000). Uprooting the tree of life. *Sci. Am.* 282: 90–95.
- Dutta, A. and Chaudhuri, K. (2010). Analysis of tRNA composition and folding in psychrophilic, mesophilic and thermophilic genomes: indications for thermal adaptation. *FEMS Microbiol. Lett.* 305: 100–108.
- Efron, B. (1979). Bootstrapping methods: another look at the jackknife. *Ann. Stat.* 7: 1–26.
- Feil, E.J., Li, B.C., Aanensen, D.M. et al. (2004). eBURST: inferring patterns of evolutionary descent among clusters of related bacterial genotypes from multilocus sequence typing data. *J. Bacteriol.* 186: 1518–1530.
- Felsenstein, J. (1981). Evolutionary trees from DNA sequences: a maximum likelihood approach. *J. Mol. Evol.* 17: 368–376.
- Felsenstein, J. (1985). Confidence intervals on phylogenies: an approach using the bootstrap. *Evolution*. 39: 783–791.
- Fitch, W.M. and Margoliash, E. (1967). Construction of phylogenetic trees. *Science*. 155: 279–284.
- Gadagkar, S.R., Rosenberg, M.S., and Kumar, S. (2005). Inferring species phylogenies from multiple genes: concatenated sequence tree versus consensus gene tree. *J. Exp. Zool. B Mol. Dev. Evol.* 304: 64–74.
- Galtier, N. and Lobry, J.R. (1997). Relationships between genomic G+C content, RNA secondary structures, and optimal growth temperature in prokaryotes. *J. Mol. Evol.* 44: 632–636.
- Gardy, J.L., Johnston, J.C., Ho Sui, S.J. et al. (2011). Whole-genome sequencing and social-network analysis of a tuberculosis outbreak. *N. Engl. J. Med.* 364: 730–739.
- Gerner-Smidt, P., Hise, K., Kincaid, J. et al., and the PulseNet Taskforce (2006). PulseNet USA: a five-year update. *Foodborne Pathog. Dis.* 3: 9–19.
- Griffiths, A.J.F., Miller, J.H., Suzuki, D.T. et al. (eds.) (2000). How DNA changes affect phenotype. In: *An Introduction to Genetic Analysis*, 7e. New York, NY: W. H. Freeman.
- Gupta, R.S. and Griffiths, E. (2002). Critical issues in bacterial phylogeny. *Theor. Popul. Biol.* 61: 423–434.
- Handelsman, J. (2004). Metagenomics: application of genomics to uncultured microorganisms. *Microbiol. Mol. Biol. Rev.* 68: 669–685.
- Hasegawa, M., Kishino, H., and Yano, T. (1985). Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *J. Mol. Evol.* 22: 160–174.
- Hedden, P. (2003). The genes of the Green Revolution. *Trends Genet.* 19 (1): 5–9.
- Henikoff, S. and Henikoff, J.G. (1992). Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA*. 89: 10915–10919.
- Hopkins, A.L. and Groom, C.R. (2002). The druggable genome. *Nature Rev. Drug Discov.* 1: 727–730.
- Huelsenbeck, J.P. (1995). Performance of phylogenetic methods in simulation. *Syst. Biol.* 44: 17–48.

- Huelsenbeck, J.P., Larget, B., Miller, R.E., and Ronquist, F. (2002). Potential applications and pitfalls of Bayesian inference of phylogeny. *Syst. Biol.* 51: 673–688.
- Hughey, R., Krogh, A., Barrett, C., and Grate, L. (1996). SAM: sequence alignment and modelling software. University of California, Santa Cruz, Baskin Center for Computer Engineering and Information Sciences.
- Jukes, T.H. and Cantor, C.R. (1969). Evolution of protein molecules. In: *Mammalian Protein Metabolism* (ed. H.N. Munro), 21–123. New York, NY: Academic Press.
- Kage, U., Kumar, A., Dhokane, D. et al. (2016). Functional molecular markers for crop improvement. *Crit. Rev. Biotechnol.* 36 (5): 917–930.
- Kimura, M. (1980). A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol.* 16: 111–120.
- Kundu, S. and Ghosh, S.K. (2015). Trend of different molecular markers in the last decades for studying human migrations. *Gene.* 556 (2): 81–90.
- Lawson, F.S., Charlebois, R.L., and Dillon, J.A. (1996). Phylogenetic analysis of carbamoylphosphate synthetase genes: complex evolutionary history includes an internal duplication within a gene which can root the tree of life. *Mol. Biol. Evol.* 13: 970–977.
- Linnaeus, C. (1735). *Systema Naturae* (trans. M.S.J. Engel-Ledeboer and H. Engel. 1964. *Nieuwkoop B de Graff, Amsterdam*). Leyden, Netherlands: Johann Willem Groot.
- Liu, Y., He, Z., Appels, R., and Xia, X. (2012). Functional markers in wheat: current status and future prospects. *Theor. Appl. Genet.* 125: 1–10.
- Loceya, K.J. and Lennona, J.T. (2016). Scaling laws predict global microbial diversity. *Proc. Natl. Acad. Sci. USA.* 113: 5970–5975.
- Margos, G., Gatewood, A.G., Aanensen, D.M. et al. (2008). MLST of housekeeping genes captures geographic population structure and suggests a European origin of *Borrelia burgdorferi*. *Proc. Natl. Acad. Sci. USA.* 105: 8730–8735.
- Mora, C., Tittensor, D.P., Adl, S. et al. (2011). How many species are there on Earth and in the ocean? *PLoS Biol.* 9: e1001127.
- Moran-Gilad, J., Rokney, A., Danino, D. et al. (2017). Real-time genomic investigation underlying the public health response to a Shiga toxin-producing *Escherichia coli* O26:H11 outbreak in a nursery. *Epidemiol. Infect.* 145 (14): 2998–3006.
- Muir, P., Li, S., Lou, S. et al. (2016). The real cost of sequencing: scaling computation to keep pace with data generation. *Genome Biol.* 17: 53.
- NCBI Resource Coordinators (2016). Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.* 44 (Database issue): D7–D19.
- Needleman, S.B. and Wunsch, C.D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* 48: 443–453.
- Parks, D.H., Porter, M., Churcher, S. et al. (2009). GenGIS: a geospatial information system for genomic data. *Genome Res.* 19: 1896–1904.
- Parks, D.H., Mankowski, T., Zangoeei, S. et al. (2013). GenGIS 2: geospatial analysis of traditional and genetic biodiversity, with new gradient algorithms and an extensible plugin framework. *PLoS One.* 8: e69885.
- Planck Collaboration (2015). Planck 2015 results. XIII. Cosmological parameters. *Astron. Astrophys. Rev.* 594: A13.
- Robinson, E.R., Walker, T.M., and Pallen, M.J. (2013). Genomics and outbreak investigation: from sequence to consequence. *Genome Med.* 5: 36.
- Ruggiero, M.A., Gordon, D.P., Orrell, T.M. et al. (2015). A higher level classification of all living organisms. *PLoS One.* 10: e0119248.
- Rzhetsky, A. and Nei, M. (1992). Statistical properties of the ordinary least-squares, generalized least-squares, and minimum-evolution methods of phylogenetic inference. *J. Mol. Evol.* 35 (4): 367–375.
- Sahraeian, S.M., Luo, K.R., and Brenner, S.E. (2015). SIFTER search: a web server for accurate phylogeny-based protein function prediction. *Nucleic Acids Res.* 43: W141–W147.

- Saitou, N. and Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.* 4: 406–425.
- Sakharkar, M.K., Sakharkar, K.R., and Pervaiz, S. (2007). Druggability of human disease genes. *Int. J. Biochem. Cell Biol.* 39 (6): 1156–1164.
- Salipante, S.J. and Hall, B.G. (2011). Inadequacies of minimum spanning trees in molecular epidemiology. *J. Clin. Microbiol.* 49: 3568–3575.
- Schmedes, S.E., Sajantilaa, A., and Budowle, B. (2016). Expansion of microbial forensics. *J. Clin. Microbiol.* 54: 1964–1974.
- Schmitt, M. (2003). Willi Hennig and the rise of cladistics. In: *The New Panorama of Animal Evolution* (eds. A. Legakis, S. Sfenthourakis, R. Polymeni and M. Thessalou-Legaki), 369–379. Moscow, Russia: Pensoft Publishers.
- Schwartz, R. and Schäffer, A.A. (2017). The evolution of tumour phylogenetics: principles and practice. *Nat. Rev. Genet.* 18 (4): 213–229.
- Searls, D.B. (2003). Pharmacophylogenomics: genes, evolution and drug targets. *Nat. Rev. Drug Discov.* 2: 613–623.
- Singapore Zika Study Group (2017). Outbreak of Zika virus infection in Singapore: an epidemiological, entomological, virological, and clinical analysis. *Lancet Infect. Dis.* 17: 813–821.
- Sokal, R. and Michener, C. (1958). A statistical method for evaluating systematic relationships. *Univ. Kans. Sci. Bull.* 38: 1409–1438.
- Strimmer, K. and von Haeseler, A. (1996). Quartet puzzling: a quartet maximum likelihood method for reconstructing tree topologies. *Mol. Biol. Evol.* 13: 964–969.
- Sturk-Andreaggi, K., Peck, M.A., Boysen, C. et al. (2017). AQME: a forensic mitochondrial DNA analysis tool for next-generation sequencing data. *Forensic Sci. Int. Genet.* 31: 189–197.
- Swofford, D.L., Olsen, G.J., Waddell, P.J., and Hillis, D.M. (1996). Phylogenetic inference. In: *Molecular Systematics* (eds. D.M. Hillis, C. Moritz and B.K. Mable), 407–514. Sunderland, MA: Sinauer Associates.
- Tamura, K. (1992). Estimation of the number of nucleotide substitutions when there are strong transition-transversion and G+C-content biases. *Mol. Biol. Evol.* 9: 678–687.
- Tamura, K. and Nei, M. (1993). Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. *Mol. Biol. Evol.* 10: 512–526.
- Tang, P., Croxen, M.A., Hasan, M.R. et al. (2017). Infection control in the new age of genomic epidemiology. *Am. J. Infect. Control.* 45: 170–179.
- Tavaré, S. (1986). Some probabilistic and statistical problems in the analysis of DNA sequences. *Lectures on Mathematics in the Life Sciences.* 17: 57–86.
- Theurey, P. and Pizzo, P. (2018). The aging mitochondria. *Genes.* 9 (1): 22.
- Thompson, J.D., Higgins, D.G., and Gibson, T.J. (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: 4673–4680.
- Wacey, D., Kilburn, M.R., Saunders, M. et al. (2011). Microfossils of sulphur-metabolizing cells in 3.4-billion-year-old rocks of Western Australia. *Nat. Geosci.* 4: 698–702.
- Weiss, M.C., Sousa, F.L., Mrnjavac, N. et al. (2016). The physiology and habitat of the last universal common ancestor. *Nat. Microbiol.* 1: 16116.
- Whittaker, R.H. (1969). New concepts of kingdoms of organisms. *Science.* 163: 150–160.
- Wilde, S.A., Valley, J.W., Peck, W.H., and Graham, C.M. (2001). Evidence from detrital zircons for the existence of continental crust and oceans on the Earth 4.4 Gyr ago. *Nature.* 409: 175–178.
- Woese, C.R. and Fox, G.E. (1977). Phylogenetic structure of the prokaryotic domain: the primary kingdoms. *Proc. Natl. Acad. Sci. USA* 74: 5088–5090.
- Woese, C.R., Kandler, O., and Wheelis, M.L. (1990). Towards a natural system of organisms: proposal for the domains Archaea, Bacteria, and Eucarya. *Proc. Natl. Acad. Sci. USA* 87: 4576–4579.
- Yang, Z. (1994). Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. *J. Mol. Evol.* 39: 306–314.

- Yang, R. and Keim, P. (2012). Microbial forensics: a powerful tool for pursuing bioterrorism perpetrators and the need for an international database. *J. Bioterr. Biodef.* S3: 007.
- Yang, Z. and Rannala, B. (1997). Bayesian phylogenetic inference using DNA sequences: a Markov chain Monte Carlo method. *Mol. Biol. Evol.* 14: 717–724.
- Yang, I.S., Lee, H.Y., Yang, W.I., and Shin, K.J. (2013). mtDNAprofler: a web application for the nomenclature and comparison of human mitochondrial DNA sequences. *J. Forensic Sci.* 58 (4): 972–980.
- Yoshida, C.E., Kruczkiewicz, P., Laing, C.R. et al. (2016). The Salmonella In Silico Typing Resource (SISTR): an open web-accessible tool for rapidly typing and subtyping draft salmonella genome assemblies. *PLoS One* 11 (1): e0147101.
- Zuckerkandl, E. and Pauling, L. (1965). Molecules as documents of evolutionary history. *J. Theor. Biol.* 8: 357–366.

10

Expression Analysis

Marieke L. Kuijjer, Joseph N. Paulson, and John Quackenbush

Introduction

The sequencing of the human genome in 2003 gave us a preliminary catalog of all human genes (Lander et al. 2001; Venter et al. 2001). Although the genome (and the collection of genes encoded within it) has evolved significantly since that first draft sequence, many questions about how gene expression is regulated and how the resulting data can be used to characterize distinct phenotypic dates and explore their properties still remain. Indeed, we know that, within a single individual, the same genome manifests itself distinctly in each and every cell type and those gene expression profiles change between conditions, including health and disease.

Scientists recognized the importance of these questions even before the genome was sequenced and developed methods for assaying how RNA expression differed between phenotypes. Although early techniques allowed only one or a small number of genes to be tested, the emergence of DNA microarray technologies opened the door to test large numbers of genes, enabling the analysis of genes across the entire genome (Schena et al. 1995). DNA microarrays were widely used to explore patterns of gene expression in model organisms and human disease (DeRisi et al. 1996; Spellman et al. 1998; Golub et al. 1999; Perou et al. 1999; Callow et al. 2000; Konstantinopoulos et al. 2011).

The early days of gene expression analysis with microarrays produced significant challenges, and many early studies were fraught with problems of irreproducibility (Ioannidis et al. 2009; Ishmael et al. 2009). However, a significant investment by computational and experimental biologists resulted in laboratory and analytical procedures that led to improved consistency in results emerging from DNA microarray studies, emphasizing the need for careful experimental design and replication throughout (Hegde et al. 2000; Simon et al. 2002; Irizarry et al. 2003, 2005; Bolstad et al. 2004; Larkin et al. 2005; Quackenbush 2005). The introduction of ultra-high-throughput sequencing technologies opened the door to RNA sequencing (RNA-seq) experiments that were far less constrained by preconceived notions about what one might measure (Kahvejian et al. 2008; Nagalakshmi et al. 2008). Despite using a very different approach to assaying expression, the development of robust RNA-seq analysis techniques built upon many of the same basic lessons learned during the development of DNA microarray analysis technologies.

The goal of this chapter is to provide a step-by-step introduction to considerations and methods for gene expression assessment, starting with experimental design and moving through questions of data normalization, comparison, and interpretation. Although introductory, we hope that the material presented here will serve as the starting point for future investigations and a more thorough examination of the methods we present.

Step 0: Choose an Expression Analysis Technology

This may seem like an unusual place to start given that RNA-seq is the dominant technology, but DNA microarrays are still widely used and have some advantages over RNA-seq that may be worth considering when developing an experimental and analytical plan.

The application of both techniques begins with the extraction and purification of RNA from samples of interest and the conversion of those RNAs to complementary DNA (cDNA) through the use of reverse transcriptase, an enzyme derived from a retrovirus. The cDNA is then used to determine, either through sequencing or hybridization, the relative abundance of genes within the genome. These abundance levels are then used in downstream analysis to understand how patterns of gene expression change between biological states and how those changes help us understand the biology of the systems being studied. While the available technologies share a common foundation, there are differences between them that are worth considering.

DNA Microarrays

DNA microarrays were the first technology developed that allowed genome-wide analysis of gene expression. DNA microarrays rely on detection of hybridization events occurring between labeled cDNA targets in solution and single-stranded, gene-specific DNA probes bound to fixed locations on a solid surface. While DNA microarrays were initially plagued by noise and often found to be irreproducible, advances in the technology and analytical methods have greatly improved the quality of the data they can produce. DNA microarrays have a number of additional advantages that make them worth considering as an alternative.

First, the gene content of DNA microarrays is well defined, such that each gene or transcript being tested for expression is represented by one or more probes (or probe sets). This has advantages in that we understand, ahead of time, which genes will be represented; therefore, we can generally determine with a reasonable degree of confidence whether a particular gene is expressed and at what relative level. While there may be cross-hybridization or other artifacts such as differential hybridization efficiencies, the quality of commercial arrays and the robustness of today's laboratory protocols have greatly improved the quality of the assays and their reproducibility.

Second, because the technology is mature, there are robust, well-established analytical methods for almost every aspect of microarray analysis. The largest single repository for DNA microarray analytical tools is Bioconductor, where there are countless, well-established methods for every aspect of microarray analysis.

Finally, there are extensive repositories of DNA microarray data available through the Gene Expression Omnibus (GEO) and ArrayExpress databases. These databases provide additional independent datasets that can be used for estimating required sample sizes to validate the findings from individual experiments.

While Bioconductor does include many methods for RNA-seq data analysis, there is less consensus as to best practices than there is with microarray analysis. Also, while GEO and ArrayExpress include RNA-seq data, those data only make up a small fraction of the total data volume found within these two resources. Although one can compare microarray data with RNA-seq data to determine general trends, there is no way to compare microarray hybridization intensities directly with RNA-seq read counts.

RNA-seq

RNA-seq can trace its roots back to cDNA sequencing in the 1990s and serial analysis of gene expression (SAGE), a technique that allowed sequencing of short cDNA fragments, in the early 2000s. However, RNA-seq really developed into its own unique approach over the last decade with the advent of ultra-high-throughput sequencing, allowing for the generation of tens of millions of sequence reads (or more) starting from minuscule quantities of RNA.

RNA-seq has quickly become the dominant technology for gene expression profiling for a number of reasons.

First, RNA-seq allows not only the expression levels of “genes” to be measured, but it can also give information on the expression levels of individual alleles and transcript variants. While this flexibility makes the technology unique and open ended, the truth is that very few studies have taken advantage of this capability, with most analyses falling back on looking at the total expression levels of the transcripts. The most significant counterexample to this is the use of RNA-seq to identify fusion transcripts in cancer – something that would be impossible to do with DNA microarrays.

Second, RNA-seq methods have dramatically improved over time, allowing smaller quantities of RNA to be used as input material. RNA-seq applications include transcript profiling from individual cells, which requires specialized analytical methods to deal with the sparsity of the data (see Single-Cell Sequencing). The use of small quantities of starting material also means that one can use small biopsy samples or multiple distinct data types (such as RNA-seq and DNA methylation data) from a single biological sample, making integrated analysis feasible.

Finally, RNA-seq is not limited by a pre-defined set of transcripts. RNA-seq experiments can uncover the expression of “new” genes that have not previously been described, including the transcript levels of non-coding RNAs. RNA-seq data can also be analyzed to detect polyadenylated viral transcript sequences.

The Choice is Yours

Although microarrays remain a viable alternative, the cost differential between microarrays and RNA-seq has fallen to the point that RNA-seq is typically the default. Given this, we will emphasize RNA-seq analyses within this chapter, referring the reader to previous versions of this book if interested in methods for microarray expression analysis. However, many of the general principles of expression analysis are the same and can be used as a general template for thinking about diverse large-scale genomic studies, and so some examples are included in our discussion below.

Step 1: Design the Experiment

In biology, most successful experiments are designed around well-established ideas of hypothesis testing. We begin by identifying a problem and postulating a mechanism. We then design an experiment in which we perturb the system in a manner that tests the hypothesis, and we then collect data that allow us to look for changes that are consistent with our postulated mechanism. The response that we observe in the system either validates or invalidates our hypothesis. In such experiments, we attempt to tightly control the variables so as to carefully measure their influence, perturbing just a single parameter at a time. Good experimental design requires sufficient replication to estimate the effects we wish to measure.

Genome-wide gene expression technologies have changed the way in which we can approach biological questions. Rather than looking at single genes, we can now survey the responses of thousands of genes in a particular system and look for altered patterns of expression that are associated with changes in phenotype. We can use these large-scale experiments to either test hypotheses or generate new hypotheses based on changes in patterns of gene expression that can later be tested. However, the scope and scale of observations enabled by genome-wide technologies do not mean we can ignore the need to carefully design experiments and analyze the resulting data.

Like all experiments, a gene expression profiling experiment should begin with a well-defined question and the experiment should collect the data necessary to answer that question. The most common designs for experiments include comparison of two experimental

groups (or *cohorts*), such as a treatment vs. a control group or a diseased vs. a healthy population. One critical element of designing such a study is assuring that the experiment has a sufficient number of independent biological replicates so that the treatment and control groups are of sufficient size to make reasonable comparisons.

Power size calculations are notoriously difficult in large-scale transcriptional profiling experiments, in large part because expression levels are so variable and relative effect sizes are generally unknown prior to conducting an experiment. One strategy that can work is to do a small pilot experiment to identify a potential signal that can be used to estimate effect size and then to use that for doing a more rigorous power calculation and designing a full experiment.

An alternate strategy is to think beyond the original experiment, including a validation stage in the experiment that uses an independent technology (such as reverse transcription polymerase chain reaction) to validate a small “significant gene set” or, better yet, including a validation population that will be independently profiled to assess whether the original results were valid.

Another important consideration is designing an experimental strategy that avoids confounding and eliminates batch effects. This includes both the experimental strategy that is used to collect samples as well as the strategy that is used to collect the gene expression data. This should include assuring that “treatment” and “control” samples are collected together and under the same conditions and that samples are mixed when RNA is collected, libraries are prepared, and sequence data are generated.

An important, and often overlooked, question is whether there are sufficient metadata on the samples that will be analyzed. For example, if analyzing samples from breast cancer, it is important to know the disease subtype of each sample and to have considered subtype distribution in the experimental design. Without such data and given what are small sample sizes relative to the number of genes being tested in an RNA-seq experiment, it is relatively easy to end up in a situation in which expression differences are the result of some bias in how samples are assigned to different groups.

For example, we previously analyzed a gene signature that claimed to predict lung metastasis in breast cancer based on expression in the primary tumor, only to discover that all of the samples with metastasis that were used to identify this signature were of the basal subtype (which is the subtype most likely to metastasize to lung). This signature was a predictor of the basal subtype, but not necessarily of metastasis. So, before analyzing the data, one needs to consider whether there could be demographic differences between treatment and control populations, or differences in the treatments experienced by patients within different subgroups. Did the patients come from different hospitals or countries? Were the patient samples collected and processed in different ways? Believe it or not, all of these confounding factors have been identified in published studies, and all of these confounding factors could have been easily avoided. It is well worth the effort to try to identify potential confounding factors *before* running the experiment, rather than trying to explain them away while analyzing the data that were collected.

One approach that we have found to be extremely useful is to begin with the analytical strategy you will use once the data are collected and to work backward to the experimental design, ensuring that you have the requisite number of samples and the appropriate metadata to assure that you have the appropriate data and information to answer your experimental question.

Step 2: Collect and Manage the Data – and Metadata

A transcriptional profiling experiment involves introducing a perturbation to a control system, collecting the biological specimens, and then generating the data that will ultimately be analyzed. While it may seem obvious that one must collect and manage the relevant data, this is an element that is often overlooked and can come back to haunt those who neglect it.

There are many ways to approach this problem, from simply storing the data in a folder on a shared drive or creating a database into which the data are ultimately placed. Regardless of the strategy one chooses, the single most important thing to do is to be organized and to document which data are associated with each project.

Step 3: Data Pre-Processing

Before data can be compared between experimental groups or used for any other purpose, one must first map the raw data to specific genes or gene transcripts. Although this might seem rather trivial, different approaches can be used – and, of course, these different approaches can potentially lead to different final results. While there are many accepted methods for performing this data pre-processing step, one should take note of and carefully document one's choices in identifying gene transcripts from raw data.

For DNA microarrays, mapping raw data might seem trivial, since one thinks of an array of consisting of fixed probes for each gene profiled. However, many arrays – most notably, Affymetrix GeneChip – use groups of probes or “probe sets” that together are used to define the expression of a gene. In fact, the Affymetrix chip design includes not only sets of “perfect match” (PM) probes designed using the reference gene sequence, but also “mismatch” (MM) probes that differ from the reference by a single base change in the middle of the probe sequence. The PM probes provide an estimate of the hybridization signal, while the MM probes are included to provide estimates of the background signal due to non-specific hybridization and background fluorescence.

The mapping of probes to genes is typically contained in a “chip design file” (CDF) that is included as input into the early stages of any analysis to provide a map between fluorescence intensity and gene expression levels. There has been considerable debate in the research community regarding what data should be used to perform this probe mapping, with some advocating using only the PM probes, others creating non-standard CDFs, and the majority using the Affymetrix-supplied CDFs. As with many aspects of gene expression analysis, there is no right answer; one just needs to make a rational choice and document that decision so others can reproduce the analysis. However, one must always be aware of what gene identifiers (and what release versions of these identifiers) the probes are mapped to – whether it be official gene names, RefSeq IDs, Ensembl IDs, or something else – as these decisions can influence downstream analyses involving the mapping of expression data to biological pathways or functional classification systems, such as Gene Ontology (GO), or the use of techniques such as gene set enrichment analysis.

RNA-seq faces a similar set of challenges, although here the mapping is somewhat less mysterious. The raw output from RNA-seq is a set of sequence reads that is mapped to a set of genes or gene transcripts. To do this, the most common approaches perform an “assembly on reference,” first mapping reads to gene transcripts, then assembling them and quantifying the overall representation for each gene. Here, the choice of reference database defines the mapping. One can choose RefSeq, Ensembl genes, or any other suitable reference. There are a host of algorithms that have been developed to map, assemble, and quantify the reads, including the Burrows–Wheeler aligner (BWA) (Li and Durbin 2009), Bowtie/Bowtie2 (Langmead et al. 2009; Langmead and Salzberg 2012), and STAR aligner (Dobin et al. 2013).

More recently methods have been developed to deal with ever larger RNA-seq datasets by using pseudo-alignment and quasi-mapping; these methods include Salmon (Patro et al. 2017), Sailfish (Patro et al. 2014), and Kallisto (Bray et al. 2016). These methods are designed specifically to attenuate the computational complexities introduced with large data, including memory restrictions. Pseudo-alignment and quasi-mapping bypass the use of a full alignment by representing the transcriptome with k -mers and mapping those to either a de Bruijn graph

representation (a graphical representation of overlaps between, and maps across, k -mers) or suffix array (a sorted array of extensions, or suffixes, of a k -mer) using a hash table. Appropriately defining hash functions allows for ignoring the majority of the reference and mapping read queries to a limited number of potential targets.

As with many aspects of genomic data analysis, there is no clear consensus as to the optimal choice, and methods are constantly evolving. What is important is to select from among the standard methods, to apply it consistently to the data that you wish to analyze, and to document your choices in a way that assures the analysis can be reproduced, including documenting software and database versions.

Step 4: Quality Control

Any measurement we make as scientists includes errors. Some of these errors are random, and statistical methods of analysis are designed to estimate the true signal given natural variation. Some errors are systematic, and these too can be estimated and handled using statistical methods. However, some errors arise from failed assays, and the best approach is to identify and eliminate the data arising from these failed assays. In the course of gene expression analysis experiments, such errors arise from contaminants within RNA samples, poor quality experimental reagents, or just simple laboratory error.

One of the single most important questions to ask once you have generated your raw data is whether those data are of sufficient quality to move through your analysis pipeline. While biological variability is something that you want to assure is represented in any dataset, failed experiments should, quite simply, be removed from the datasets being analyzed. Expression analysis in the laboratory involves many complex steps, and anything from degraded input RNA to bad reagents to simple mistakes can produce data that are dominated by such high levels of noise that they can derail any sort of meaningful analysis. Fortunately, there are a host of tools that can be used to analyze data generated by both microarray expression analysis and RNA-seq experiments to provide well-established sets of metrics for both microarrays and sequence-based data. As is true with everything in this field, the tools used to analyze these data will continue to evolve rapidly, so the reader is encouraged to keep abreast of literature reviews or to reach out to colleagues actively performing gene expression analyses regarding new approaches that may have come to the fore.

Quality Control Tools

The Bioconductor package `arrayQualityMetrics` provides a wide range of tools (including many assembled from other Bioconductor packages) for assessing the quality of both single-color and two-color microarray data. As input to the `arrayQualityMetrics` package, one provides a matrix of microarray intensities and, optionally, information about the samples and the probes in a Bioconductor object of class `AffyBatch`, `ExpressionSet`, `NChannelSet`, or `BeadLevelList`, which are all objects that coordinate expression data from different technologies with phenotype.

The output from `arrayQualityMetrics` includes a false-color representation of each array and an MA plot to assess its quality. In an MA plot, the M value is the log-ratio of two intensities and the A value is the mean of the logarithm of the intensities. For two-color arrays, these plots use intensities from each channel and, for single-color arrays, the value of M uses the median intensity of each sample as the denominator in the ratio. An example of an MA plot on data before and after normalization is shown in Figure 10.1, where the systematic curvature below the horizontal axis is removed by the normalization process.

There are also a number of other diagnostic plots that can be used to identify bad single arrays or overall bad datasets. These include the RNA degradation plot from the `affy` package (Gautier et al. 2004), the relative log expression (RLE) boxplots and the normalized unscaled

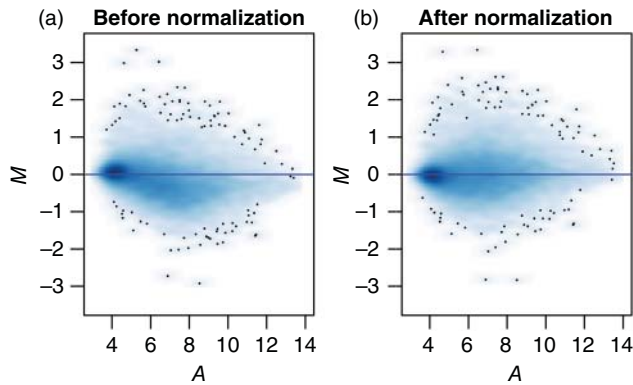


Figure 10.1 Example of an *MA* plot before (a) and after (b) normalization. *A*, or the mean of the log-transformed expression levels, is shown on the *x*-axis; *M*, or the log-ratio of the expression levels in the sample of interest and the median intensity of the expression levels of the probes across each sample, is shown on the *y*-axis. *M* vs. *A* is drawn for each probe in the expression dataset using the “smoothScatter” option in the function “ma. plot” from R package “affy.” In correctly normalized data, we expect these points, on average, not to deviate from the horizontal blue line. In the figure made on the non-normalized data, we see a slight downwards trend, which is removed after normalization.

standard error (NUSE) boxplots from the *affyPLM* package (Brettschneider et al. 2008), and the QC stat plot from the *simpleaffy* package (Wilson and Miller 2005). The results obtained with these quality control tools are assembled into an HTML document that provides a valuable resource for understanding the raw data you have assembled.

For RNA-seq, *FastQC* is a widely used package that provides a collection of simple tools for quality control checks on raw high-throughput sequence data in a manner very similar to the microarray *arrayQualityMetrics* package. *FastQC* has a number of analytical modules that allow users to explore various aspects of sequence quality, providing a number of summary graphs and tables and exporting the results to an HTML-based report. Within *FastQC* are modules that provide basic statistics, including data on the number of reads, the read length, and GC content. Users can also view box-and-whisker plots showing per base assessment of sequence quality scores at each position along all reads. One can also get a plot of the distribution of per sequence quality scores. Both of these provide a good overall assessment of the quality of the sequence run.

Another useful plot to assess overall sequence quality is the per base sequence content. One would expect that, for any genome, the GC content should be consistent along the length of any random sequence read, with %A = %T and %G = %C. However, library preparation protocols generally ligate short primer and adapter sequences to the 5' end of the DNA to be sequenced, and this is where one would expect to see substantial deviations in GC distributions. A related measure is the per base N content, which quantifies how often a defined nucleotide has been substituted for an N because of the inability to call a base with sufficient confidence; this information can help identify failed cycles in the sequencing reaction.

An example of a histogram of the base mismatch (MM) rate, relative to a reference sequence, for a set of samples on which RNA-seq was run is shown in Figure 10.2. While most samples have a low MM rate, there are a few outliers that could be removed from downstream analysis. There are also tools to identify aberrant levels of sequence duplication, over-represented sequences, missed adapters, and over-represented *k*-mers. *FastQC* also has tools for analysis of microRNAs, metagenomic sequences, and epigenetic assays such as methyl-seq.

One exciting new tool that provides an overview of a study's quality is *MultiQC*. The tool aggregates quality control reports on multiple samples from *FastQC*, as well as other tools, and presents these in a single HTML report that is easy to read and digest and that can help identify and subsequently remove poor quality samples from the analysis.

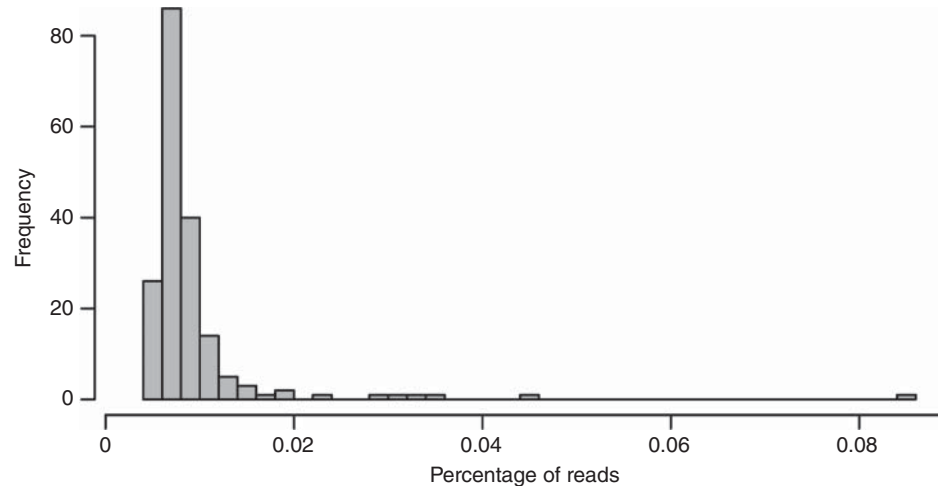


Figure 10.2 Histogram of the base mismatch (MM) rate across multiple RNA-seq samples. Most of these samples had a base MM rate <0.04%. One could decide to remove outlier samples based on the distribution of the base MM rate.

Screening for Misidentified Samples: PCA on Y Chromosome Expression

One element we have not touched on is the quality of the annotation associated with each sample. Any analysis will ultimately rely on assigning samples to different groups, comparing expression levels between groups, and ensuring that there are no confounding factors that might skew the analysis. The quality of the analysis (and the confidence we have in our conclusions) depends on the reliability of how individuals are assigned to particular groups; this, in turn, depends on whether we can accurately associate each sample with appropriate annotation metadata, such as an individual's sex, age, treatment status, and other phenotypic data. While this may seem relatively trivial, mis-annotation of samples is a far more frequent problem than one might expect. For example, 46% of studies available in GEO have been found to have poor or incorrect annotations – errors that could have been easily identified prior to submission by employing simple validation steps (Toker et al. 2016).

It is generally difficult to test for accuracy in sample annotation, as the purpose of most experiments is typically to find differences between groups rather than to use known differences to assign samples to groups. That said, there is one test that can be run on virtually any dataset that can give us some sense of the quality of the sample annotation: whether males and females are annotated correctly. If one simply looks at the expression of Y chromosome genes and performs principal component analysis (PCA; see Principal Component Analysis), one would expect to find two distinct groups, as females do not express Y chromosome genes (Paulson et al. 2017).

As an example, we analyzed colorectal cancer gene expression data from The Cancer Genome Atlas (TCGA; from the Genomic Data Commons [GDC] Data Portal) and five colorectal cancer datasets from GEO (GSE14333, GSE17538, GSE33113, GSE37892, and GSE39582). When we used PCA (see Principal Component Analysis) to analyze expression of Y chromosome genes, we found two distinct clusters of samples – one expressing those genes (and therefore likely male) and a second with only expression at the level of background noise (and therefore likely female). However, we found that 11 of 456 samples (2%) in the TCGA were misidentified by sex, meaning that samples annotated as female grouped with the males (thus expressing Y chromosome genes) and samples annotated as female grouped with the males. However, when we looked at the GEO datasets, we found 85 of 1376 (6%) samples misclustered by sex. Although we could remove the misidentified samples from downstream analyses, we dropped an entire GEO study because the mis-annotation rate was nearly 15%, leading us to question the veracity of the remaining sample annotation.

Step 5: Normalization and Batch Effects

The Importance of Normalizing and Batch-Correcting Data

The output from any gene expression analysis can be represented as an expression matrix populated by positive values that represent the observed expression levels for each probe or gene in each sample. One can represent these data as an “expression matrix” or, for RNA-seq data, as a “count matrix” (C), where each row is a gene, each column is a sample, and the entry at each location is the observed number of reads mapped to that gene in that particular sample.

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{bmatrix}$$

In a perfect world, one could directly compare the expression levels between any group of samples by simply comparing the expression levels or read counts gene by gene between those samples. However, there are many factors that can skew those data, including unequal amounts of starting RNA, artifacts in library preparation, differential efficiency in hybridization or sequencing, or a host of other issues.

Normalization is a common procedure in data analysis that allows us to make comparisons between different samples or different datasets. For example, in northern blot analysis, one commonly uses housekeeping genes such as actin or GAPDH to adjust hybridization signals of other genes in each sample, as a way to compensate for variation in sample quantities in the experiment. In this case, the assumption is that one or more genes are expressed at the same level in all samples, and that one can use this “baseline” to adjust the measurements of other genes. Broadly, there are a few types of normalization processes that can enable comparisons between samples or genes.

FPKM and Count Data

In a DNA array, each gene is typically represented by probes that are roughly equivalent to the probes used for every other gene. For example, on an Affymetrix array, genes are represented by probe sets. Each probe in a probe set is 25 bp in length and has corresponding PM and MM probes differing at a single base (exactly in the middle of the probe). In addition, each probe set contains the same number of probes and is located near the 3' end of its target transcript so as to help assure near uniform efficiency in preparing hybridization libraries from the RNA.

Considerations regarding per gene normalization are somewhat different when analyzing RNA-seq data. Here, libraries are prepared and sequenced and, if one simply counts the number of sequence reads per gene, larger genes are more likely to pick up more reads than shorter genes. Consequently, raw count data are often transformed to either reads per kilobase million (RPKM), fragments per kilobase million (FPKM), or transcripts per million (TPM) values. The three measures differ subtly in how they are calculated, but all start by mapping reads to transcripts and then scaling the results.

To determine RPKM, one simply counts the number of reads mapping to a transcript and divides this by the number of reads per million that map to the genome, normalized by the length of that gene. This last step is to account for the fact that twice as many reads will map to a 2 kb gene than to a 1 kb gene. For example, if 4 million reads map to a genome, and 5000 reads map to a particular gene that is 2 kb in length, the RPKM value for that gene would be 625 ($[5000/2]/4$). FPKM is an extension of RPKM and is used when performing paired-end sequencing, where both ends of an RNA-seq library fragment are sequenced. The concept underlying FPKM is identical to that for RPKM, taking into account the fact that two reads can map to the same fragment.

TPM values are similar to those discussed above but are normalized to a standard “per million” value that can more easily be compared between samples. To calculate TPM, one starts

with each gene and divides the number of reads (or transcripts) by the length of the gene in kilobases, providing a reads per kilobase (RPK) value for each gene. The RPK values for all of the genes in the genome are then added together to calculate a cumulative RPK value; this value is then divided by 1 million to obtain a “scaling value.” Finally, each gene’s RPK value is divided by the scaling value to calculate a TPM measure for each gene. This TPM is conceptually closest to the microarray measure in that it takes into account the length of each gene and then compares these normalized transcript counts.

Sample and Quantile Normalization

As more and more analyses were being performed on high-throughput gene expression data generated using DNA microarrays, it quickly became apparent that the assumption that there were “invariant” housekeeping genes was simply not correct and that, in fact, *all* genes varied in their expression levels. Lacking a solid reference, the focus of new normalization techniques shifted to looking at the distribution of gene expression levels across all genes in a sample, then adjusting the distributions to be similar to each other.

At first, normalization methods adjusting the mean or median expression levels for a sample were used, but these methods failed to compensate for differences in distributions that may be due to experimental artifacts. If one assumes that a cell can only make a certain quantity of RNA, then one would expect that, as some genes increase expression, other genes must decrease their expression levels such that the distribution of expression levels is the same for related samples.

Conceptually, one easy way to do this is to look at the distribution of gene expression levels and slice it into smaller segments, or *quantiles*. One then can adjust the data, quantile by quantile, so that all the samples in an experiment have the same distribution and so that changes in the expression of any particular gene can be compared between samples. Conceptually, this sounds relatively simple, but it is worthwhile looking at (and understanding) the process in a bit more detail.

The process of quantile normalization is depicted in Figure 10.3 using an example that involves four samples and six genes. We represent these measures in a genes-by-samples matrix and use three simple procedures to normalize the data. First, following the blue arrows in the figure, one takes each gene and calculates a median value across all of the samples. These

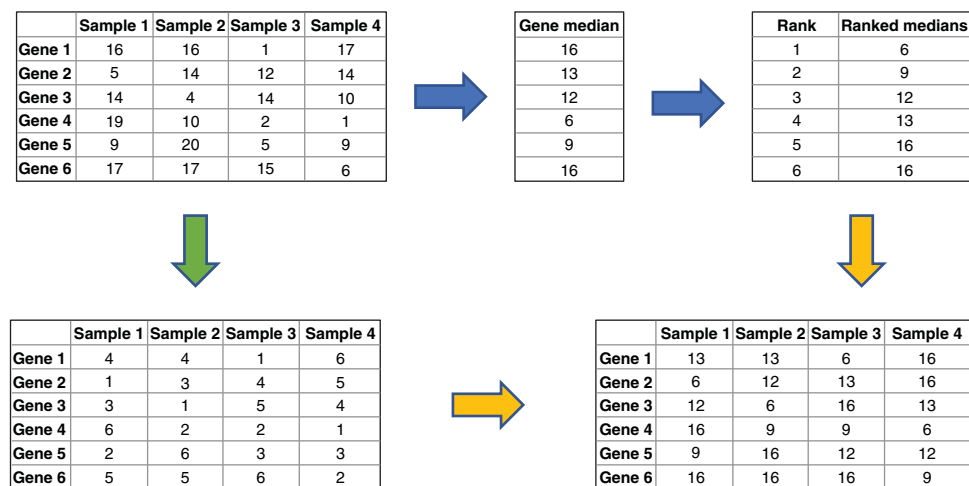


Figure 10.3 Overview of quantile normalization. We start with the box on the top left, which includes expression data for four samples and six genes. To quantile normalize the data, we first calculate the median expression of each gene and rank them from lowest to highest (blue arrows). We then rank the genes in each sample based on their expression levels (lowest to highest; green arrow). Finally, we replace the ranks in the rank matrix with the values that correspond to the same ranks in the ranked medians (gold arrows).

median values are then ranked from lowest to highest. Second, following the green arrows in the figure, one looks at each sample being analyzed and replaces the expression value of each gene with its rank from lowest to highest in that sample, producing a ranked matrix. Finally, following both of the gold arrows in the figure, one combines the rank matrix and the ranked set of medians, replacing the ranks in the rank matrix by the corresponding ranked median values.

In the example in the figure, all rank 1 genes are assigned an expression level of 6, all rank 2 genes are assigned an expression level of 9, and so on, so that the original values are now replaced by the ranked medians. An important assumption behind this approach is that the reference is appropriate for all samples, which may not be true when the underlying biology is different (e.g. when analyzing multiple tissue types). While this process might seem counter-intuitive, this method has been shown to produce robust and reproducible expression values that can be compared across samples (Bolstad et al. 2003).

Additional Methods of Sample Normalization

The choice of normalization methods is one of the greatest sources of contention in almost any discussion of gene expression analysis, given that each method relies on some set of assumptions that might be more or less reasonable to anyone examining a particular dataset. Below, we discuss four additional methods that have been widely used (Li et al. 2015).

Counts per Million The counts per million (CPM) scaling method is similar to TPM in that the count values are normalized to a standard “per million” value that can more easily be compared between samples. CPM and other scaling methods are based on the assumption that each cell can generate more or less the same amount of RNA such that the total number of reads, or counts, should be constant. To calculate CPM, one simply totals the reads for each sample and then scales all sample read counts so that they are equal. CPM, TPM, and gene-length normalized values were among the most widely used normalization methods, particularly in the early days of gene expression analysis based on RNA-seq.

Upper Quantile Normalization This scaling method assumes that count distributions are similar at the low- to mid-expression levels but deviate from each other above the 75th quantile. This method simply scales each dataset such that the numbers of counts below the 75th quantile are set equal to each other across samples and the normalized count, y_{ij} , is scaled such that:

$$y_{ij} = c_{ij}/q_{75j}.$$

Relative Log Expression This method assumes that count values closely follow the geometric mean of gene expression values across samples and that read count frequencies increase exponentially with sequencing depth. RLE uses the geometric mean for each sample and scales the reads in each sample so that the geometric means are the same. A normalizing factor is calculated for each sample as the median of the ratio between feature read counts and the geometric mean of read counts across all samples. This approach is used as the standard normalization method by DESeq (Anders and Huber 2010) and DESeq2 (Love et al. 2014), which are described below.

Trimmed Mean of M Values The trimmed mean of M values (TMM) approach is based on the assumption that the majority of genes are not differentially expressed. Here, TMM makes use of a *single* sample as the reference. The method then compares each sample with the reference, calculates log fold-changes (what is often referred to as the “ M ” value in microarray analysis), removes the outer 30% of M values, and calculates an average M_0 value which is

then scaled to be equal for all samples. TMM is used within the edgeR (Robinson et al. 2010) testing framework (edgeR is described below).

Batch Correction

Batch correction is another important aspect of any large-scale genomic analysis – and, for that matter, any scientific study that collects measurements in groups or of different samples at different times. Batch effects are systematic sources of error that can be introduced because measurements are made, for example, using different reagents, under differing laboratory conditions, using similar samples collected at different times, and given the inherent variation in how different people conduct the same assay. Given the large number of measurements made on each sample, batch effects are particularly evident in high-throughput experiments.

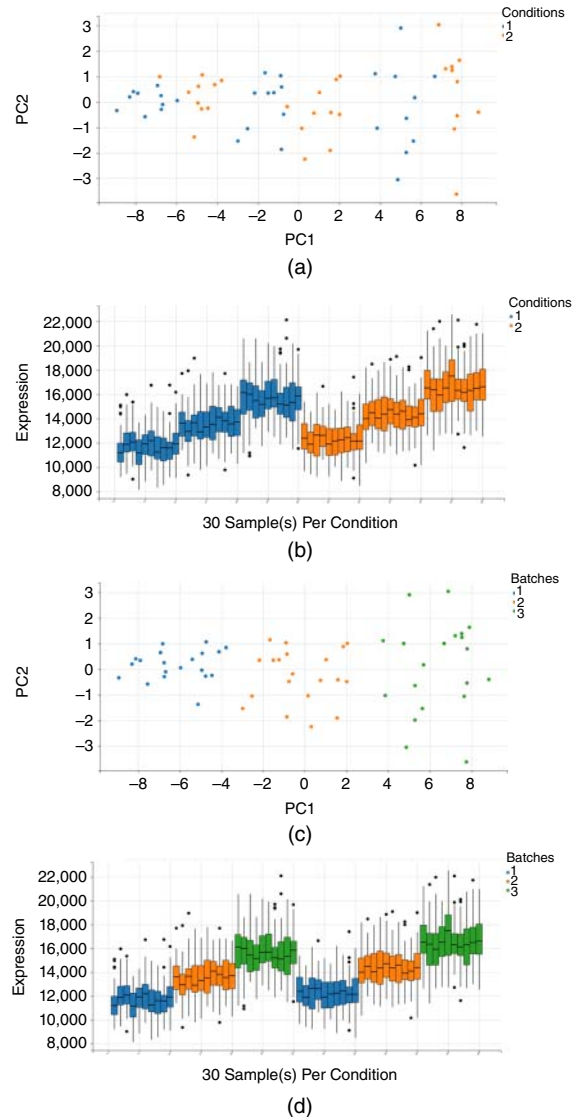
The first line of defense against batch effects is good experimental design. One should make every effort to collect all samples at the same time, then assay them at the same time under the same conditions and using the same set of reagents whenever possible. While this is generally not practical for large numbers of samples, the next best thing is to mix cases and controls at each step so that sample groups are not confounded by batch effects. For example, if one were to measure gene expression in controls on Monday and cases on Tuesday, then any batch effect that might occur would be confounded with case/control status, making it impossible to resolve differences. Here, mixing cases and controls would correct for any underlying batch effects.

A very easy way to test for batch effects is to generate plots from a PCA (see Principal Component Analysis) based on gene expression data so that each point in the plot represents a sample. One could then color the points in this plot by condition (for example, cases and controls). In a perfect world, one would expect to see a clear separation between conditions, but what you see will depend on the signal and noise and potential batch effects. Next, one can recolor the plot based on other relevant variables such as sample collection date, RNA extraction date, labeling or library construction date, array batch or sequencer run, and so on, looking for patterns. In an ideal world, there will be no patterns in the PCA plot except for a separation based on condition. However, patterns often appear that group samples from one batch together, or separate different batches from each other. It is these differences that batch effect corrections attempt to control for. In a more high-throughput fashion, highlighting the correlation between the PCA components and phenotypic and batch variables is often a useful technique. An example that walks the reader through this process is shown in Figure 10.4.

It should be noted that, although extremely useful, PCA generally only captures large batch effects. Indeed, individual genes or gene subsets can be affected adversely by experimental conditions, producing effects that might alter the conclusions of any downstream analysis. Fortunately, methods have been developed that can address batch artifacts, identifying experimental signals that are correlated with batches and correcting for those to better allow identification of differentially expressed genes that are associated with experimental groups.

Two widely used methods for batch correction are COMBAT (Johnson et al. 2007) and surrogate variable analysis (SVA; Leek et al. 2012). Both of these examine the expression data and look for genes whose expression is correlated with batch, or, in the case of SVA, other non-biological variables. However, and as described above, these associations can only be found and corrected for if the relevant biological variables are not confounded with these non-biological variables. For example, if all cases are run in one batch and all controls in the next, batch effects will confound the phenotypes; if there are differences that are due to some non-biological factor, the batch effects will cause these differences to appear as real biological differences. Finding and removing batch effects relies on having all experimental groups represented, to the degree possible, in each of the laboratory batches.

Figure 10.4 Batch effects principal components analysis (PCA) example. Boxplots and scatterplots of data simulated using the BatchQC software. Data were simulated using BatchQC, per the vignette. (a) A PCA scatterplot of the first two principal components where points represent samples and are colored by phenotype condition. (b) Boxplots of the simulated gene expression for multiple genes within each sample are also highlighted and colored by phenotype condition. (c, d) The same data; however, each sample is colored by sequencing “batch” and a large difference can be observed in the first principal component and gene expression values by batch.



Step 6: Exploratory Data Analysis

A commonly asked question when analyzing large-scale genomic data is whether there are subgroups within a population that are defined by distinct patterns of gene expression. This is a question that can only be reasonably answered when there are enough data available to search for patterns that can be used to identify and distinguish groups. The methods that are used for such unbiased searches are called *unsupervised*, as the searches find patterns rather than asking if there are patterns that can distinguish pre-defined groups. There are many methods that fall into this broad general class of methods, but the most commonly used ones are hierarchical clustering, PCA, and non-negative matrix factorization (NMF).

As noted previously, a convenient way to represent transcript data is using an expression matrix – a genes-by-samples matrix in which each row is a “gene vector” that represents the expression levels for a particular gene across all samples and in which each column is a “sample vector” representing the expression levels for all genes in a single sample; each element in the matrix represents a single gene in a single sample. A *heat map* is a representation of

that matrix in which each cell in the heat map is colored based on the intensity of its signal. Both hierarchical clustering and NMF group subsets of samples and/or genes based on shared patterns of expression and visualize results in the context of a heat map. PCA performs operations on sample or gene expression vectors – the aforementioned columns or rows of the matrix.

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 visualized easily (Eisen et al. 1998; Michaels et al. 1998; Wen et al. 1998). Initially, one starts with N clusters, where N is the number of samples (or genes) in the target dataset. Hierarchical clustering is an agglomerative approach in which single expression profiles are joined to form nodes; these nodes are further joined until the process has been carried to completion, forming a single hierarchical tree.

Hierarchical clustering essentially asks which vectors are closest to each other, then groups samples together based on their distance from each other. Of course, there are many ways of measuring distance between samples (or genes) based on their expression profiles; among the most common are the Euclidean distance measure (which works well when the absolute level of gene expression is important) and the Pearson correlation distance measure (which is best when correlated patterns are important).

Having chosen a method for measuring distance (Figure 10.5), the algorithm proceeds in a straightforward manner. The ensuing description assumes that samples are being grouped, although the clustering process works in an identical fashion for genes.

- 1) Calculate the pairwise distance matrix for all of the samples to be clustered.
- 2) Search the distance matrix for the two most similar samples or clusters. Initially, each cluster consists of a single sample. If several pairs share the exact same similarity score, one typically chooses one pair at random, although other methods can be used to decide between those pairs.
- 3) The two selected clusters are merged to produce a new cluster that now contains two or more objects.
- 4) The distances are calculated between this new cluster and all other clusters. There is no need to recalculate all distances as only those involving the new cluster have changed.
- 5) Steps 2–4 of this list are repeated until all objects are in one cluster.

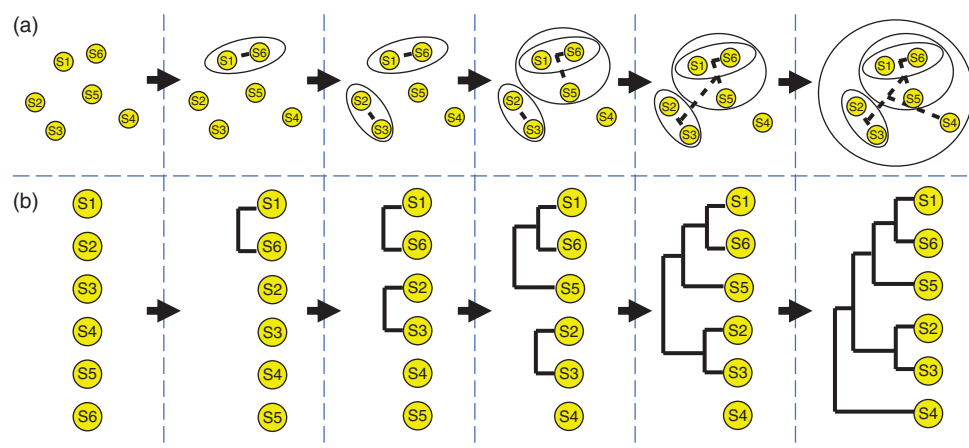


Figure 10.5 A simple illustration of the process of hierarchical clustering. (a) The repeated search for samples, or groups of samples that are “close” to each other and their subsequent merging to form larger clusters. (b) The corresponding formation of a hierarchical clustering dendrogram, joining samples together based on their assessed similarity.

There are a number of variations of the hierarchical clustering method that reflect different approaches to calculating distances between the newly defined clusters and the other genes or clusters (what are referred to as *agglomeration* methods):

- single linkage clustering uses the shortest distance between one cluster and any other
- complete linkage clustering takes the largest distance between any two clusters
- average linkage clustering uses the average distance between two clusters.

Typically, the relationship between samples is represented using a dendrogram, where branches in the tree are built based on the connections determined between clusters as the algorithm progresses. To visualize the relationships between samples, the dendrogram is used to rearrange the rows (or columns as appropriate) in the expression matrix heat map to visualize patterns in the dataset (Figure 10.6).

The tree-line structure of the dendrogram makes it useful for identifying places where one might divide the samples into some number of clusters simply based on their appearance. However, rather than using the “eyeball test,” it is better to use an objective method for determining the number of clusters and their membership – and, fortunately, there are multiple ways to search for such groups.

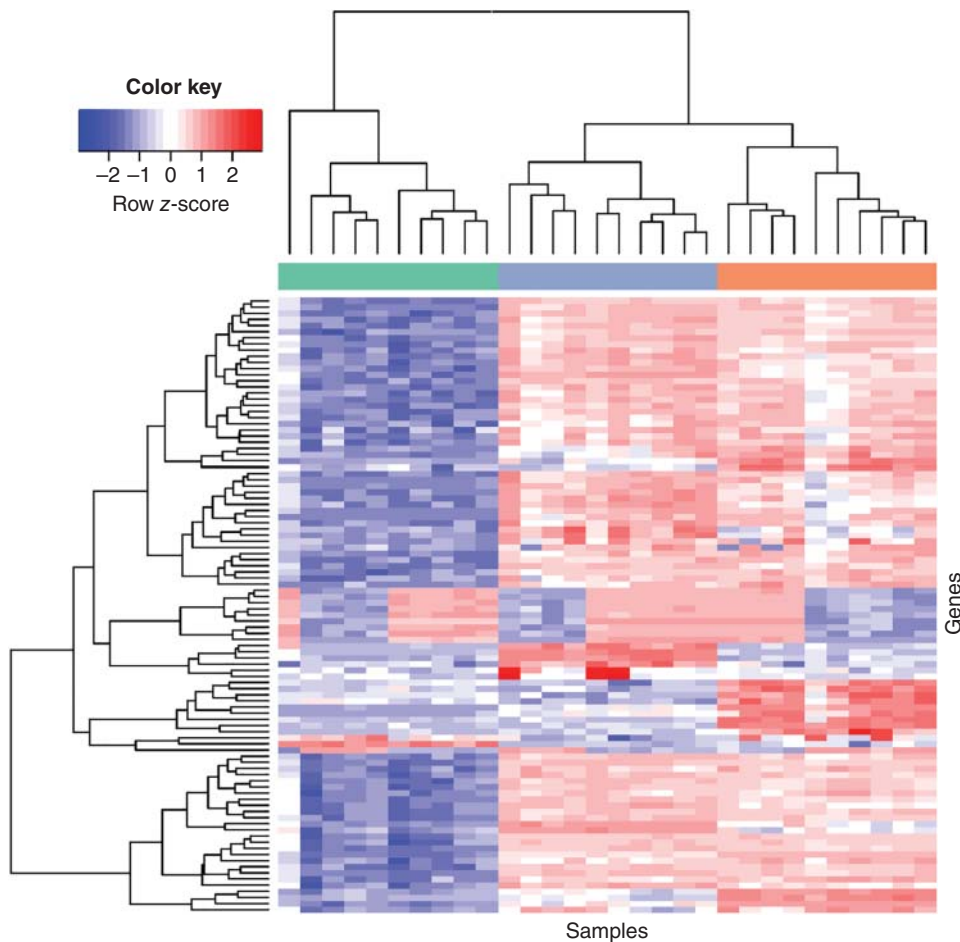


Figure 10.6 Heatmap showing clustering of gene expression data of the 100 most variable genes in three different heart tissues. The expression levels in this heatmap are row z-score normalized to best show the differences in expression. Low expression is visualized in blue and high expression in red. The dendrogram on top is obtained by performing hierarchical clustering using Euclidean distance. It shows that samples from the three tissues (top color bar) cluster into three separate groups. The tissue on the left (green) has low expression of these genes, while the center tissue (gray) and the tissue on the right (orange) have higher expression of these genes. A small subset of the genes is highly expressed in the tissue on the right, but is lowly expressed in the other two tissues.

One method simply is to use the distances calculated in building the clusters as a measure of the connectivity of the individual clusters. As one moves up the dendrogram from the individual elements, the distance between clusters increases. Consequently, as one increases the distance threshold, the effective number of clusters decreases. An alternative approach is to use bootstrapping or jack-knife techniques to measure the stability of relationships in the dendrogram, using this stability as a measure of the number of clusters represented. There are a number of bootstrapping approaches that can be used, but the simplest is to perform sampling of the dataset with replacement, each time calculating a new hierarchical clustering dendrogram and simply counting how often each branch in the dendrogram is recovered; a percentage cut-off on the dendrogram sets the number of clusters. In making a bootstrap estimate for gene cluster stability, it is appropriate to resample the collection of biological samples, whereas in estimating the number of clusters in the biological samples one bootstraps the gene expression vectors. Jack-knifing is similar, but, instead of resampling, the appropriate vectors are sequentially left out as new dendrograms are calculated, continuing until all vectors have been considered. Again, the stability of each cluster is estimated based on how often a given relationship in the dendrogram is recovered.

One potential problem with many hierarchical clustering methods is that, as clusters grow, the expression vector that represents the cluster when calculating distances may no longer accurately represent any of the elements within the cluster. For example, in clustering genes, the “center” of each cluster is typically an average over all of the genes within that cluster; the resulting linear combination of gene expression vectors is sometimes referred to as a “metagene.” Consequently, as clustering progresses, the actual expression patterns of the genes themselves become less relevant. Furthermore, if a bad clustering assignment is made early in the process, that error is fixed in place and cannot be corrected. An alternative that can avoid these artifacts is to use a divisive clustering approach, such as *k*-means, to partition data (either genes or samples) into groups having similar expression patterns.

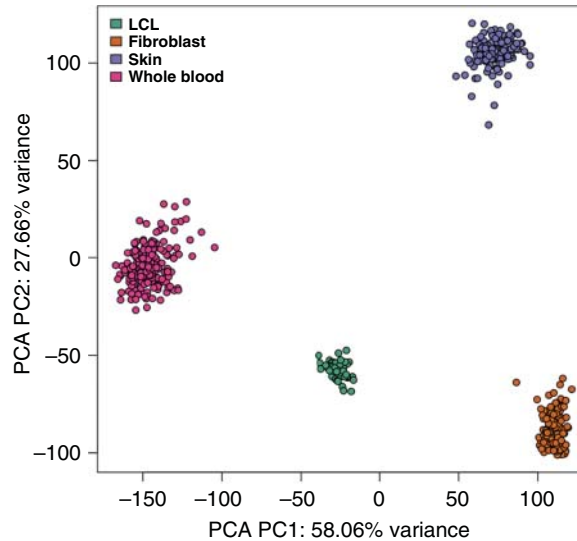
Although clustering approaches work with any dataset, in practice they often do not work well for large datasets in which many of the genes do not vary between samples. Consequently, it can be useful to first apply a statistical filter to the data, selecting only those genes that are the most variable between experimental classes. However, this filtering can lead to biases in the clustering. If one chooses genes that distinguish two experimental groups, then the most likely result of the filtering will be two clusters in which the samples fall into the two pre-defined experimental groups. A more unbiased approach is to simply calculate the variance of each gene across the population of samples and then eliminate those genes that are not changing significantly in the dataset, as these genes are the least likely to shed any light on subclasses that exist in the sample collection. However, this latter approach relies on having a good balance of samples across experimental groups.

Principal Component Analysis

If we look at a samples-by-genes expression matrix, we can imagine that each sample has its own unique expression level for each of the 25 000 (or so) genes being evaluated. We can then represent each sample as a point in that 25 000-dimensional “gene expression state space.” As you might imagine, the collection of samples could be visualized as a cloud of points within that 25 000-dimensional space. However, many of those genes are likely correlated in their expression levels and, therefore, do not provide much information that can be used to separate samples and to distinguish different groups of samples within the cloud.

PCA (Figure 10.7) is a dimensionality reduction method that searches for linear combinations of variables – in this case, the expression levels of genes – that best explain the variance between the samples, and then transforms the data such that the eigenvectors of gene expression (an eigenvector of a linear transformation is a non-zero vector of which all values change by the same scalar factor when that transformation is applied to it) are ranked to best separate the data. In this way, the data are transformed such that the first “component” explains the largest amount of variation in the dataset, the second component explains the next largest

Figure 10.7 First two components of principal component analysis (PCA) on the normalized gene expression matrix for skin, whole blood, and cell lines derived from those tissues (data obtained from the Genotype Tissue-Expression project (GTEx) v6). Each point represents a sample and is colored by its source. The first principal component (PC1) separates tissue types, the second component (PC2) separates tissues from cell lines.



amount of variation, and so on. Plotting the data using these eigenvector coordinates generally makes it easier to visualize the separation of samples into distinct groups. This, in turn, can help in understanding whether the samples in an expression dataset group into specific subsets with large differences in gene expression. PCA is also a good quality control tool, as technical variation such as batch effects can easily be detected by visually inspecting the PCA plots.

PCA is based on a number of simple linear algebra transformations on the underlying genes-by-samples expression matrix. A schematic overview of how PCA works is shown in Figure 10.8.

- 1) Begin by standardizing the matrix (in this case, the rows of the matrix) such that the range of expression of each gene is on the same scale.
- 2) Calculate the covariance matrix, where the entry ij is the covariance between gene i and gene j . The covariance between two genes basically measures whether they are correlated in their deviation from average expression of all the samples in the population.
- 3) Calculate the eigenvectors and eigenvalues of the covariance matrix. An eigenvector is a vector that, when multiplied by the covariance matrix, returns the same vector with each value multiplied by a scale factor (the corresponding eigenvalue). The eigenvectors, or principal components, are invariants of the matrix and are linear combinations of the genes (and so are sometimes referred to as “eigengenes”).
- 4) Use the eigenvectors to recast the original data. This is accomplished by simply multiplying the original expression matrix by the matrix of eigenvectors.
- 5) Plot the results in the basis of the new eigenvectors (which are orthogonal to each other, much like the x - y - z axes).

In this framework, the first eigenvector explains the greatest amount of variation in the data. The second eigenvector the second greatest amount of variation, and so on. In two- or three-dimensional plots, it is common to examine the distributions, coloring the samples by, for example, batch, the sex of the subject, or treatment group, to see how various systematic and biological factors influence variation in the data.

Non-Negative Matrix Factorization

NMF is another dimensionality reduction method. It models gene expression data as a product of two non-negative matrices by summarizing genes into a smaller number of so-called “meta-genes.” In NMF, we start with an $n \times m$ (genes-by-samples) expression matrix R . We

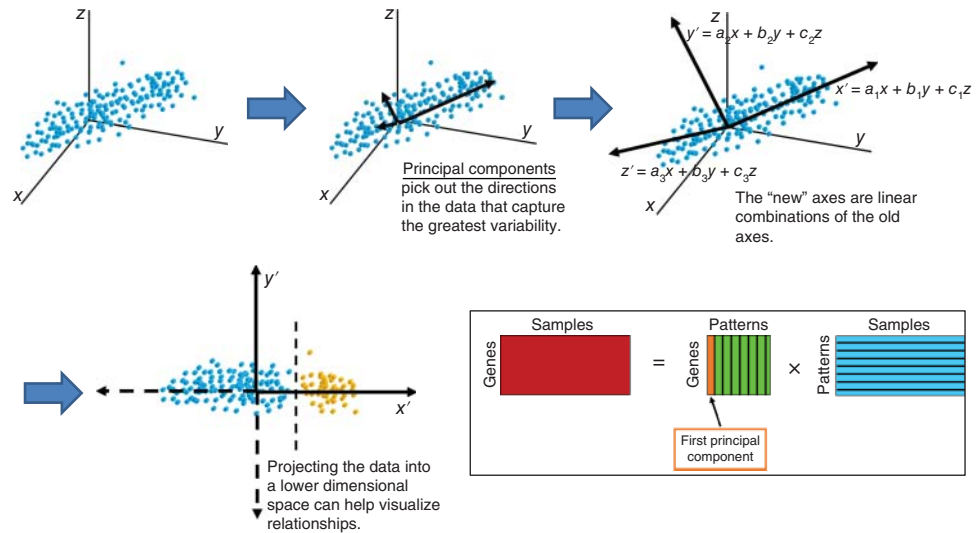


Figure 10.8 Principal component analysis (PCA) is a dimensionality reduction method that identifies combinations of variables that capture the greatest variation in the data, and then plots the data in principal component space. Here, points represent experiments in a higher dimensional “expression space” in which each sample has its own unique expression profile (and therefore unique coordinates). PCA identifies orthogonal axes along which the data have the greatest variation and calculates new coordinate axes that are linear combinations of the individual genes. The samples are then projected into “PC space,” where, typically, only the first principal components are plotted. Mathematically, PCA decomposes our genes-by-samples matrix into a genes-by-patterns matrix (the columns of which are the principal components) and a patterns-by-samples matrix. The principal components in this case are sometimes referred to as “metagenes” since they consist of linear combinations of genes.

use an $n \times k$ features matrix P that has the centroid values for every gene in each of k clusters. We multiply P by an $n \times k$ weights matrix Q that provides weights for representing columns of R as non-negative linear combinations of the columns of P . The resulting product \hat{R} is an approximation of the original matrix R :

$$R \approx P \times Q^T = \hat{R}$$

Hidden in this explanation is the fact that NMF requires some advanced knowledge of how many experimental groups (k) one might expect in the data.

In practice, one often does not know how many experimental groups to expect in any dataset, so a common practice is to run the method with multiple values of k and then choose the partitioning of the data that best explains the biology of the system under study. A quantitative measure that is useful in such an exploratory analysis is the cophenetic coefficient, which measures how similar genes have to be such that they are grouped into the same cluster – essentially, a measure of within-to-between cluster distance. A widely used approach is to plot the cophenetic coefficient and then look for a precipitous drop in its value (indicating that likely true clusters are being fragmented) to choose the optimal k . An example of such a plot is visualized in Figure 10.9.

Step 7: Differential Expression Analysis

While exploratory data analysis can be very useful, most gene expression experiments are designed to test the hypothesis that differences in phenotype are associated with differences in the expression of functionally relevant genes. The most straightforward approach to testing this hypothesis is to ask whether there are genes that have significantly different levels of gene expression between the sample groups.

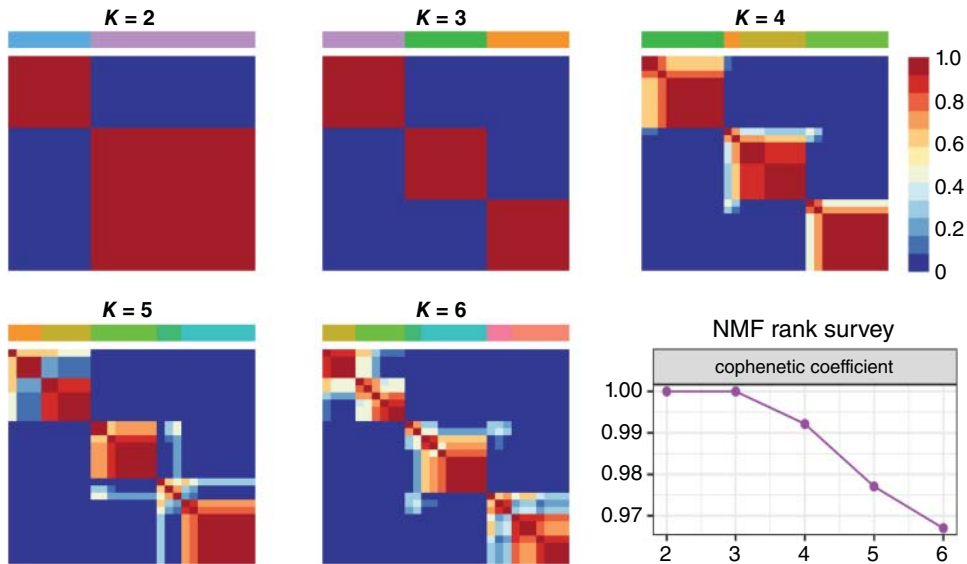


Figure 10.9 Illustration of how one can select k when performing consensus clustering with non-negative matrix factorization (NMF). The heatmaps show the consensus clustering based on different k values, ranging from 2 to 6. The color bars on top of the heatmaps show how the samples are divided into 2–6 groups. The color bar on the right shows the consensus value, which ranges from 0 to 1, with 1 meaning that samples have similar expression levels. In this example, as k becomes higher, the clusters become less precise. The “NMF rank survey” shows the cophenetic coefficient (y -axis) for each k . As can be seen, at $k = 2$ and $k = 3$ the cophenetic coefficient is high, but at higher k it becomes lower. Generally, one selects k based on the highest k before a breakpoint in the cophenetic coefficient plot. In this example, that would be $k = 3$. Therefore, this example dataset likely contains three different subtypes. This figure was generated using R package “nmf.”

The earliest applications in gene expression analysis simply applied a biological filter, looking for genes that changed, on average, by a factor of 2 or more between conditions (a fold-change filter). While an intuitive measure to many biologists, it ignored both the magnitude and variability of gene expression. As a result, statistical measures of differential expression quickly became the standard for assessing transcriptional differences.

The biggest problem with using statistical methods in gene expression analyses is that there are tens of thousands of genes on which we have data, but typically only a few tens or hundreds of samples. This leads to the problem of multiple testing – the observation that, with so many more measurements than samples, it is likely that some genes will differ between sample groups simply by chance. Fortunately, there are ways to correct for such problems. We will start by examining the various methods that are used to test for significant differences as these methods, and the assumptions on which they are based, can help in understanding how we can best identify significant differences.

Student’s t -Test: The Father of Them All

Student’s t -test, or more simply the t -test, is the most widely used method to determine differences between two groups in any field. The t -test can be characterized as a measure of signal to noise in that it compares mean expression levels between two groups, then uses the standard deviation to determine whether the difference in means is significant. Essentially, the test measures whether the difference in means is large compared with the variation in the data and estimates the probability that the observed difference is due to chance.

The t -test comes in different flavors, such as a two-sided t -test (testing whether the expression of a gene is higher or lower in one group than in another), a one-sided t -test (used to test whether expression of one gene is higher in one group than another), or a paired t -test (used to test whether the difference in expression of a gene between groups is larger than one might

expect). There is also a generalization of the t -test for use with more than two groups called the F -test or analysis of variance (ANOVA).

The paired t -test is the most widely used in expression analysis when one has matched samples, such as patients and matched controls. In its native form, the t -test assumes that the data have a normal distribution, so it does not correct for any potential mean-variance dependency that can occur in gene expression studies. One possible way to overcome some of these problems is to use an empirical t -test that permutes samples between groups and calculates repeated t -statistics for each gene, testing whether the observed (actual) t -statistic with real data is larger than what the permuted data tell you to expect.

However, all versions of the t -test suffer from the problem of multiple testing, which means that, with many, many more genes being profiled than samples to constrain them, we risk finding genes that are different between groups simply by chance. Consequently, there has been a substantial investment of time and effort in developing more robust methods for identifying genes that are differentially expressed between experimental groups. Here, as before, we survey widely used methods with the recognition that there are many other methods that have been developed and used in the analysis of expression data.

Limma

The linear models for microarray and RNA-seq data (Limma) approach was, as its acronym implies, first developed for the analysis of microarray data. Limma fits a linear model to the expression levels for each gene and uses a moderated t -test to identify significantly differentially expressed genes. The moderated t -test is constructed using an empirical Bayesian method that differs from a standard t -test in that the variance is scaled based on expression levels, using a pooled estimate of variance and degrees of freedom to better estimate significance; this provides a stable estimate of significance even for small sample sizes. The Limma method also calculates an estimate of false discovery rates (FDRs) rather than a simple p value, as a measure of significance; this is described in more detail in False Discovery Rate.

Voom

The negative binomial model is one of several statistical methods developed for use with count data, but calculating a negative binomial distribution can be computationally difficult for large numbers of samples, making methods like DESeq and edgeR impractical. While log transformation of RNA-seq counts can help standardize the data, it often further skews variance estimates. Variance modeling at the observational level (or Voom) empirically models the mean-variance association for each gene, fitting the standard deviation of each gene's logarithm of the counts per million (log-cpm) value as a function of the average log count. Voom then incorporates the mean-variance trend as a precision weight for each observation and uses this within the Limma analysis pipeline. This has the advantage of providing an empirical Bayes framework for linear modeling of RNA-seq data and then allows simple integration of Voom into established pipelines for analysis. Voom's compatibility with established methods has made it one of the most widely used methods for RNA-seq analysis.

Negative Binomial Models

Differential expression analysis for sequence count data (DESeq) (Anders and Huber 2010; Love et al. 2014) and edgeR (Robinson et al. 2010) are two widely used methods for identifying differentially expressed genes from RNA-seq analysis methods. Both methods use the RNA-seq count data which suffer from over-dispersion (meaning more variability than would be expected for unbiased count data). Both DESeq and edgeR decompose the variance of a gene's expression level into both biological variability and technical variability. Biological variability contributes to differential expression between genes. The technical variability results from various aspects of the experimental process, including errors introduced in sample collection, RNA extraction, library preparation, sequencing, and other factors in the experiment,

all of which can distort the actual RNA counts. Both methods assume that the amount of biological variability is directly proportional to the amount of over-dispersion – the increased variability seen in a dataset compared with what would be expected using solely unbiased count data. Both DESeq/DESeq2 and edgeR assume a negative binomial distribution (rather than a Poisson distribution) and fit a generalized linear model to estimate and account for this over-dispersion when identifying differentially expressed genes.

Fold-Change

As mentioned previously, measurements of fold-change were widely used in the earliest days of DNA microarray genome-wide expression analysis. While statistical methods give an unbiased estimate of the evidence supporting the differential expression for a particular gene, they ignore the intuition of many biologists that small changes in the expression level of a particular gene should not be associated with large differences in phenotype. While one might argue that small changes in the level of a transcription factor, a kinase, or some other protein might have larger downstream effects, it can be difficult to interpret small absolute perturbations for most genes. Consequently, many studies use a combination of fold-change and statistical significance, filtering the statistically significant genes to include only those with a fold-change greater than 2 (or some other threshold) in any downstream analysis.

Correcting for Multiple Testing

As noted previously, the large number of genes assayed in any RNA-seq experiment and the relatively small number of samples increases the likelihood that one finds differentially expressed genes simply by chance. For example, imagine analyzing 25 000 genes and ranking them by some measure of significant difference between experimental groups (such as their *t*-statistic). Each individual *t*-statistic represents an individual test on a single gene, and, with 25 000 genes, that many individual tests are performed. If you were to take the top 5% of the genes based on your statistical measure, you would have selected 1250 genes without being able to say with confidence that any one of these is truly differentially expressed. Fortunately, there are methods for dealing with this multiple testing problem.

Family-Wise Error Rate The family-wise error rate (FWER) is a way of estimating the probability of making one or more false discoveries (false positives or type I errors; see Chapter 18) when performing multiple statistical tests. If we are performing some number *c* of tests that have a pre-defined level of significance that we want to use for each test (α), then we can calculate the FWER as

$$\text{FWER} \leq 1 - (1 - \alpha)^c$$

If we return to our example of 25 000 genes and an estimated level of significance of 0.05, then we would expect (FWER probability close to 1) to have at least one false discovery, as computed by

$$\text{FWER} \leq 1 - (3.8 \times 10^{-55})$$

The Bonferroni correction is a well-established FWER method for dealing with the multiple testing problem. One simply divides the *p* value threshold by the numbers of tests that are performed, replacing α with (α/c) . However, this is known to be too stringent a criterion for gene expression analyses; with 25 000 genes, a *p* value cut-off of $p < 0.05$ is reduced to $p < 2 \times 10^{-6}$, a threshold so low that most analyses fail to find even a single gene that meets this threshold, even when comparing samples that are distinctly different. While there are adjustments to the Bonferroni correction that can help mitigate its severity, the most widely used methods are based on other measures that estimate FDRs.

False Discovery Rate Benjamini and Hochberg (1995) introduced the FDR concept as a way of dealing with statistical issues arising from multiple testing, an idea which was later extended by Benjamini and Yekutieli (2001). While FWER estimates the probability of one or more false positives, FDR recognizes that false positives will occur and tries to estimate the number of false positives within a group of supposedly significant results, allowing the user to select a tolerated proportion of significant results likely to be false positives. The FDR is essentially the proportion of significant results that one would expect to be false positives. Note that the FDR discussed here (which is specific to multiple testing and is formally an “FDR-controlling procedure”) is similar but fundamentally different than the FDR discussed elsewhere for binary classification (see Box 5.4).

In its simplest implementation, calculating the FDR from the p values for any statistical test is relatively easy. If we assume that we have N tests, and we calculate p values for each test, then to calculate the FDR we perform the following steps.

- 1) Sort the individual p values from smallest to largest: $p_1, p_2, \dots, p_k, \dots, p_N$.
- 2) For an FDR of q , search for the k th p value such that

$$p_k \leq (i/N)(q/c(N)), \text{ where } c(N) = \sum_{i=1}^N (1/i)$$

One then declares as significant (at an FDR of q) those genes whose p values rank in the range $1, \dots, k$.

- 3) Methods that calculate FDRs often report a q value for each gene, which can be obtained from the above equation and written as

$$q_i = (p_i N / i) c(N), \text{ where, as before, } c(N) = \sum_{i=1}^N (1/i)$$

- 4) However, the q values calculated in this way are not a monotonic function of the p values, so Benjamini and Yekutieli introduced the adjusted q value, q_i , where

$$q_i = \min q_k \text{ for } k \geq i$$

Understanding FDRs and their use is extremely important for analysis of gene expression data as most methods report FDRs or q values as a default. For methods that do not report FDRs as a default, one can use functions such as the “p.adjust” function from the “stats” package in R to compute them.

Step 8: Exploring Mechanisms Through Functional Enrichment Analysis

Having identified a set of “significant” differentially expressed genes, the next step is to use this list to explore the biology of the system under study. If you have a reasonable understanding of the system you are studying, it is relatively easy to look at the significant gene list and selectively describe one or more genes and their potential involvement in the process under study. However, such “bio-poetry” means that any such interpretation is ultimately based on anecdotal knowledge and can fail to capture real trends in the data. Rather than asking about individual genes, a better approach is to look for biological processes that might be altered in their overall expression patterns between states.

Fortunately, there are many resources that can be used to provide higher order annotation for genes (see Chapter 13). Among the most widely used annotation systems is GO, a well-established and well-curated system that uses the biological literature and other sources of information to assign each gene to a classification in each of three aspects (see Chapter 7):

- 1) cellular component (CC) – the regions of a cell or its extracellular environment to which a gene’s protein product localizes
- 2) molecular function (MF) – the primary function carried out by a gene product at the molecular level, such as transport or binding

- 3) biological process (BP) – the collective process in which a gene product participates, such as cell growth, signaling, or energy metabolism.

Among these, the GO biological process information is often the most informative. In addition to GO, there are many pathway databases and other databases of “gene sets” that can be used for classification, but the methods used with these are essentially identical. What we want to know is whether there is one or more biological process (or pathway) that is over-represented among the most significant genes distinguishing the study populations. These methods are referred to as gene set enrichment analyses, or functional enrichment analyses, and many different approaches, R packages, and online tools exist. The most widely used methods are either list based, which uses a predetermined set of differentially expressed genes, or rank based, which uses the entire gene list ranked by some metric of significance, such as a *p* value or a *q* value.

List-Based Methods

Having identified a set of differentially expressed genes, the question is whether there are more genes in that list mapping to a particular functional class than one would expect by chance. Most of the methods used to make such an assessment are based on Fisher’s exact test (also referred to as the hypergeometric test, but called “exact” since it does not use an approximation to the hypergeometric function). The Fisher’s exact test relies on a selected set of genes (those that have been identified as differentially expressed) and a background set. For microarrays, the background set is the set of genes that appear on the array. For RNA-seq analyses, the background is often the whole genome.

Fisher’s exact test is based on constructing a contingency table for each functional category to be tested (for example, for each GO BP term). One maps significant and non-significant genes to that GO BP term and then tabulates the others. So, if there were *n* genes in total, *a* of which are significant and map to that BP term, *c* of which are significant but do not map to that BP term, *b* of which are not significant but map to that BP term, and *d* of which are neither significant nor map to the tested BP term, then the corresponding contingency table would be of the following form.

	Number of significant	Number of non-significant	Row total
Annotated to tested BP term	<i>a</i>	<i>b</i>	<i>a + b</i>
Annotated to other BP terms	<i>c</i>	<i>d</i>	<i>c + d</i>
Column total	<i>a + c</i>	<i>b + d</i>	<i>a + b + c + d</i>

The hypergeometric distribution is then used to estimate the probability of observing such a distribution by chance:

$$p = \frac{\binom{a+b}{a} \binom{c+d}{c}}{\binom{n}{a+c}} = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{a!b!c!d!}$$

So, if 10% of the genes in the genome are annotated to a particular BP term, there would be a high likelihood of seeing 10% of the significant set also mapping to the same BP term. However, if 20% of the significant genes mapped to that same BP term, that may well be significant. The value of using this method is that it helps guard against over-interpreting the representation of functional classes. Because there are many functional classes being tested, these *p* values must be corrected for multiple testing.

An example of a widely used list-based method for pathway enrichment analysis that is based on (a modified version of) the Fisher’s exact test is the Database for Annotation, Visualization and Integrated Discovery, or DAVID (da Huang et al. 2009a,b). DAVID’s test is modified by replacing *a* with (*a* – 1) in the contingency table, which makes the test more conservative in

estimating significance. DAVID is a user-friendly online tool that can test a given input and background gene list against different pathway annotation databases, including GO terms and pathways from Biocarta and the Kyoto Encyclopedia of Genes and Genomes (KEGG). DAVID has background gene sets for the most common microarray platforms, as well as whole genome collections for RNA-seq analysis.

An R package that performs a similar type of analysis is topGO (Alexa et al. 2006). In addition to performing a hypergeometric test, topGO can perform pathway analysis using algorithms that are specifically designed to take into account dependencies between different GO terms. An example is the elim algorithm found within topGO, which removes genes that are annotated to a significantly enriched node (or GO term) from all its ancestor nodes in the GO tree structure. In this way, less emphasis is placed on general cellular functions, with a greater emphasis on the more specific GO functional terms that appear further down in the GO hierarchical structure.

Rank-Based Methods

The problem with list-based methods is that they are sensitive to where one sets the threshold for inclusion. One often finds that adjusting the FDR cut-off for significance up or down can result in very different results because there may be a group of genes, just above or below the significance threshold, that are all from a particular functional class. Rank-based approaches get around the problem of significance thresholding by ranking all of the genes in the assay by some measure of their significance (such as a p value, q value, t -statistic, signal-to-noise metric, or some other method). The rank-based approaches then test whether genes annotated to a particular functional class (such as a GO BP term or a KEGG pathway) are over-represented near the extremes of the rank-based list. The first implementation of this approach is called Gene Set Enrichment Analysis (GSEA; Subramanian et al. 2005). GSEA is available as both an online tool and a stand-alone Java program that can be called from other programming languages, such as R. GSEA uses a weighted Kolmogorov–Smirnov test to determine a so-called enrichment score of a gene signature, then uses permutations to identify whether the enrichment score is significant (Chapter 13). The R package GSEAlm (Oron et al. 2008) uses a linear model to calculate p values and then tests whether the distribution of p values for genes mapping to a particular annotation class is different from the p value distribution for the background gene set.

Step 9: Developing a Classifier

Many analyses have the identification and exploration of biological processes driving differences in phenotypes as their ultimate goal. However, one other common application of gene expression profiling, particularly in clinical or translational applications, is to use the data to develop a classification model to assign a new sample to one of the phenotypic groups under study. To develop such a classifier, one must first select a set of features (in our case, genes) that distinguish between biological classes and then fit the parameters of some model so that it can accurately classify samples based on the expression of the feature set (called “training” the algorithm). As with the other steps in gene expression analysis outlined here, there are myriad choices that one could make in selecting features and in training and testing a classification algorithm. These include many statistical and machine learning methods that can be used for classification, but there is no clear consensus in the field as to which methods might be optimal. That said, it is clear that biomarkers, which include a feature set and a classification method, need to be carefully validated using independent datasets.

One key element in developing a successful and reproducible classifier is starting with a good experimental design. There have been thousands of gene expression biomarkers published, most of which have not been used beyond the initial study in which they appear. While these

were addressed in part in our earlier discussion about experimental design, there are a few additional criteria that should be considered.

First, there should be a good balance of the different groups one wants to classify. If you are studying a rare disease such that only 10% of your population has that disease, then you can build a classifier that is 90% accurate by simply concluding that no one has the disease. So, one should recognize this and either try to balance the groups or, better yet, clearly articulate the criteria for success.

Second, one needs to consider having both training and test populations. In the standard paradigm, one performs feature selection and parameter fitting/algorithm training on a single test set and then validates the predictive model on an independent test set (meaning a set that was not previously used to select features or to train the algorithm in any way).

If one has a small population from which to draw (for example, if dealing with a rare disease), cross-validation is an acceptable approach, but for each “fold” of the cross-validation one should re-perform both the feature selection and algorithm training using the training subset, then re-test the performance of the method on the independent test subset. The challenge with a cross-validation method is that the result is not a single classifier but, rather, a collection of classifiers, none of which can be easily validated relative to the others.

The other thing to recognize in developing classifiers is that one needs training and test datasets for which there is an objective truth; otherwise, methods cannot be effectively trained, nor can their performance be objectively assessed. One alternative way to look at this problem is to turn the test set and training set paradigm on its head by using multiple, independent training sets, learning a new classifier on each training set, and then testing the concordance and stability of multiple classifiers on a single test set. In many ways, this yields a better metric for classifier success as it speaks to how likely we are to make the same classification of an individual patient, independent of where or how the algorithm was fit.

Measuring Classifier Performance

In measuring classifier accuracy, two commonly used measures are the *sensitivity* and *specificity* (see Chapter 7). Imagine that we have developed a classifier and that we want to test it on an independent dataset for which we already know the classes of each sample. We can use the classifier method, make a determination for each sample in the test dataset, and then check how well we did. If we imagine that we have cases and controls and want to classify cases, then we will consider the cases as positives and controls as negatives. We can then classify our predictions as true positives (TPs) and true negatives (TNs), meaning that the predictions confirm the true classifications. In turn, false positives (FPs) and false negatives (FNs) represent disagreements with the true classifications. We can then define the sensitivity, or true positive rate (TPR, sometimes also called the hit rate or recall), as the proportion of TPs that were found relative to the actual number of real positives:

$$\text{TPR} = \text{TP}/(\text{TP} + \text{FN})$$

We then define the specificity, or true negative rate (TNR), as the proportion of TNs that were found relative to the actual number of real negatives:

$$\text{TNR} = \text{TN}/(\text{TN} + \text{FP})$$

A third measure that is sometimes useful is the precision, or positive predictive value (PPV), defined as the proportion of TPs that were found relative to the number of called positives:

$$\text{PPV} = \text{TP}/(\text{TP} + \text{FP})$$

Finally, a very useful diagnostic plot that incorporates these concepts is the receiver operating characteristic curve (or ROC curve, named because of its development in radar detection during World War II), where sensitivity is plotted against the specificity (or FPR). This graphical representation is quite useful, as most classification methods include parameters that can

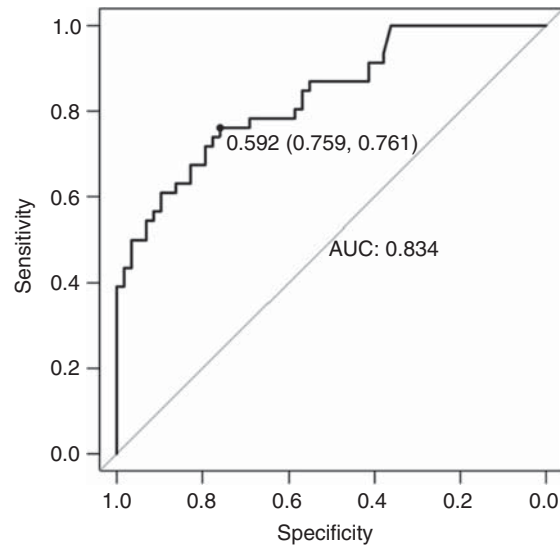


Figure 10.10 Receiver operating characteristic (ROC) curve for a model designed to use microbiome data to predict whether individuals have symptomatic diarrheal disease (Pop et al. 2016). The area under the ROC curve, or AUC, is a measure of how accurately a classifier performs, balancing sensitivity and specificity. The diagonal line (where sensitivity = specificity) represents the results expected by chance for a random classifier. A good classifier has an ROC curve that extends above the random line and consequently has an AUC > 0.5. An AUC < 0.5 would be expected for a classifier with a negative predictive power.

adjust the sensitivity or specificity; understanding how the two affect each other can help determine how to adjust the prediction model. A random classifier would have an equal TPR and FPR and would appear as a diagonal in the ROC curve (Figure 10.10).

Feature Selection

As noted previously, a classifier has two components: a feature set and a classification algorithm. The goal in feature selection is to use comparisons of sample groups in the training set to identify a set of genes that distinguish those groups and that have sufficient discriminatory power to classify new samples.

Differential Expression Testing It may not surprise you that one of the most common methods for feature selection is to use the statistical methods for differential expression analysis described in step 7. The statistical tests for differential expression analysis will identify genes that, in any given dataset, best distinguish the experimental groups. While intuitive, differential expression analysis may identify a large number of highly correlated genes that could bias any downstream classification system. What one really wants is a set of features that have the greatest discriminatory power between classes, based on the multiple patterns that might be necessary to provide full class discrimination. Fortunately, there are a number of methods that select such features, including those we describe below, such as minimum redundancy maximum relevance (mRMR) and significance analysis of prognostic signatures (SAPS).

The challenge with feature selection was highlighted by Venet et al. (2011), who compared published gene sets against random gene sets for their ability to separate breast cancer cases into groups with significant differences in survival. The ability of random gene signatures to outperform “significant” gene sets suggests that measures beyond statistical significance in selecting gene sets for classification are needed.

Minimum Redundancy Maximum Relevance Statistically significant sets of genes often have groups of genes that are highly correlated because they are co-expressed but that all represent similar processes. If you use only the most significant genes in a classifier, you run the risk of over-sampling large correlated gene sets and, in doing so, missing the range of biological processes that might help to distinguish distinct phenotypes.

Consider a case where you are comparing different classes and ranking the genes by their significance, selecting the most significant gene and then removing the genes that are highly

correlated with it from consideration, then selecting the next most significant gene, removing correlated genes, and continually repeating the process. What this should yield is a set of genes that are both highly predictive and relatively distinct from each other. As you now most likely recognize, there are many ways of measuring similarity, including the use of Pearson correlation coefficients and Euclidean distance.

A method that helps balance the representation of different biological processes when measuring similarity is called minimum redundancy maximum relevance (mRMR) (Ding and Peng 2005). mRMR uses mutual information (a non-linear method of association) to simultaneously identify genes that best discriminate between classes and reduce the potential overlap in expression profiles between genes. An implementation of mRMR in R is available in the `survcomp` package and a parallel version can be found in the `mRMRe` (De Jay et al. 2013) package.

Significance Analysis of Prognostic Signatures The paper by Venet et al. (2011) examined the ability of random gene sets to predict survival, and their findings cast doubt on many predictive methods that have been published based on selected gene sets. Part of what might underlie this finding could be the relatively large number of genes that are used in many published classifiers and correlations between the genes in the selected feature sets (and others in the genome), among other reasons. SAPS (Beck et al. 2013) is a heuristic method that can address the aforementioned considerations and is appropriate for use when attempting to determine predictors of disease outcomes or survival, although the general procedure could be adapted to other classification problems as well.

The method calculates a SAPS score for a candidate gene set based on three separate p values – P_{pure} , P_{random} , and $P_{\text{enrichment}}$ – each of which is estimated using a series of tests.

- 1) P_{pure} is calculated by first using k -means clustering (with $k = 2$) to separate patients into two groups based on the selected set of genes, then computing a log-rank p value to estimate the probability that the two groups of patient samples exhibit no difference in survival.
- 2) Next, random gene sets of the same size as the candidate gene set are selected and tested as described in step 1 of this list to assess how well they separate the population into two groups that differ in survival. P_{random} is the proportion of random gene sets that have a log-rank p value at least as significant as P_{pure} .
- 3) $P_{\text{enrichment}}$ examines the candidate gene set and the random gene sets to determine their relative enrichment for highly predictive genes based on a concordance index. The concordance index of a gene is the probability that, for a pair of patients randomly selected in the dataset, the patient whose tumor expresses that gene at a higher level has a worse outcome than the patient whose tumor expresses the gene at a lower level. $P_{\text{enrichment}}$ is calculated by using a pre-ranked gene set enrichment analysis to determine the enrichment of genes with high or low concordance indices in the candidate gene set relative to the random gene sets selected in step 2 of this list. The significance of enrichment is estimated using a permutation analysis.
- 4) Finally, these three scores, together with the direction of the association (*direction*, which is “1” for positive and “-1” for negative associations) between the candidate gene set and outcome, are combined to calculate the SAPS score:

$$\text{SAPS score} = -\log_{10} \max(P_{\text{pure}}, P_{\text{random}}, P_{\text{enrichment}}) \times \text{direction}$$

The larger the absolute value of the SAPS score, the more significant the prognostic association of all three p values. The statistical significance of the SAPS score can be estimated by permuting genes, generating a null distribution for the SAPS score, and computing the proportion of gene sets from the null distribution that have at least as large an absolute value of the SAPS score as that observed with the candidate gene set. If multiple candidate gene sets are evaluated, the raw SAPS p value of each gene set can be used to generate a corresponding SAPS q value, which is the SAPS p value corrected for multiple testing.

The value of methods such as mRMR and SAPS is that they provide a means of testing gene sets for their quality before training and testing an algorithm. Using an optimal gene set can greatly increase the chance that a classification method will perform well, although additional validation of the gene set plus the classification algorithm are essential.

Classification Methods

Having selected a candidate gene set for classification, the next step in the process is choosing, training, and validating a classification algorithm that can be used to assign a new sample to one of the phenotypic subgroups under study. There are a large number of classification methods borrowed from statistics and machine learning for this purpose. These include nearest centroid, shrunken nearest centroid, Gaussian subtype classification models, k -nearest neighbors, support vector machines, random forests, linear discriminant analysis, quadratic discriminant analysis, partial least squares, logistic regression, neural networks, and others (Hastie et al. 2001, 2009; Haibe-Kains et al. 2012) (see also Chapter 18).

While the details of each of these methods can differ substantially from the others, each classification method represents a mathematical function whose parameters are fit (“estimated” in statistics, “learned” in machine learning), whose input variables are gene expression levels for a particular sample, and whose output is a subgroup assignment for that sample. While it would be wonderful to provide guidance as to the best method for classification, a survey of the literature will demonstrate that there is no scientific consensus as to the best approach to use (although many papers claim superiority of one relative to others). There is, however, consensus that the performance of those methods should be rigorously tested and validated as described in more detail below.

One important question to consider when training a classifier is the relative cost of FPs and FNs. Most methods optimize overall performance, but there can be instances where there is a real cost for over- or under-identifying members of a class. For example, in a clinical setting, it might be far better to identify everyone who has a disease (increased sensitivity) at the risk of having some level of FP identification (decreased specificity). Depending on our application, we might decide it is important to minimize the FPR, the FN rate, the PPV, or some other parameter. In most instances, optimizations are performed with equal value placed on FPs and TPs and FNs and TNs. So, before fitting a model, it is useful to know whether this is an appropriate assumption to make, or if there are some misidentifications that are more costly than others. That decision will help guide the approach to both model fitting and validation.

Validation of Predictive Models

A predictive classifier is only useful if it is accurate and reproducible. In breast cancer, for example, there have been thousands of subtype classifiers published, but fewer than 10 are used in a clinical setting. One common reason for the failure of these predictive models is that their performance is often over-estimated because of methodological errors, a phenomenon that is known as over-fitting. There are many reasons this can occur, and we will examine three strategies that can help avoid over-fitting.

Validation of Population-Level Predictions Using Independent Test Sets The performance of most classifiers is based on their ability to partition a group of samples into subgroups that are defined by their underlying biology or some other measure like response to therapy or disease survival. If we start with a test dataset for which we have subgroup information, we can perform feature selection and then train our method, fitting model parameters. If we applied that same classifier to the test dataset, we should obtain a classification accuracy of 100%, but that is not a fair test. The appropriate question is whether this classifier, applied to a truly independent

test set, will provide a sufficient level of accuracy to be of use. Independent validation requires the following.

- 1) Starting with a training dataset, using known sample classes to perform feature selection and to fit model parameters.
- 2) Using an independent test dataset that has been completely unused in any prior study, with class labels blinded, and applying the classification model that you have developed to assign samples to each subgroup.
- 3) Unblinding the sample class labels, then comparing the predicted class with the true class, calculating metrics such as the sensitivity and specificity before reporting the findings.

This is a relatively simple and well-established protocol, but there are some things to keep in mind when using it. First, an independent dataset will give you more reliable answers if it is truly independent. If you are using clinical samples, try to have training and test samples come from different hospitals, have RNA extracted by different people, and have expression measured by different facilities. If it is possible to generate multiple test sets that are independent from the training set and from each other, then you are more likely to accurately estimate performance.

One common mistake that was made often in the early days of transcriptome profiling was to pool samples for feature selection and then to separate the pool for training and testing. The problem with this approach is that biases in some samples can influence feature selection, potentially inflating the final performance estimates. The important thing to keep in mind is that training and test sets have to be kept truly independent from each other at every step in the process.

Validation of Population-Level Predictions Using Cross-Validation A variation on the independent training and test set paradigm, and one that is often used when there are only a limited number of samples available, is cross-validation. An n -fold cross-validation uses a single dataset, dividing it into training and test sets n times, then repeating the training and test process with each of the n “folds” of the process. There is no absolute right way to divide the initial dataset or a correct number of folds to perform but, given that this approach is often used with relatively small datasets (such as for a rare disease), a common split is 90/10 in training vs. test and the use of at least 10 folds. The procedure mirrors the independent validation model.

- 1) For each fold, divide the dataset into separate training and test sets.
- 2) Use the known sample classes in the training set, perform feature selection, and fit model parameters.
- 3) Use the fit model to classify the samples in the test set.
- 4) Calculate the performance of the classifier.
- 5) Repeat steps 1–4 in this list n -fold times, tabulating an overall average performance of the classifier and its method.

Cross-validation is one of the most misused methods for classifier evaluation. One common mistake is to use the entire dataset for feature selection, and then to use the various folds to train the model and apply it. The problem here is that there may be biases in some of the samples that get captured during feature selection and then inflate the performance of the overall method. So it is essential that, within each fold, the training and test sets remain independent. Further, the entire process should be run multiple times to ensure that a particular data split does not bias the results.

The problem with cross-validation is that, at the end, there is no single classifier that you can report. Instead, there are n feature sets and n classifiers, one for each fold in the cross-validation. What some groups have done is to take the intersection (or sometimes the union) of the classification gene sets and then report those as a biomarker set. Sometimes, that consensus set is used to train the algorithm on the entire dataset, but there is no solid

theoretical support for using this method and, while one might take this approach, the overall performance of this new classifier would remain unknown until it is used with a truly independent dataset.

Validation of Individual-Level Assignment Robustness Using Independent Training Sets Validation of classifiers using a training set and one or more independent test sets provides a very good way of understanding how well a classification method will perform on a population, and the sensitivity and specificity are good measures of population-level performance. However, if you are a patient in a clinic and if the physician treating you is going to use some test for determining a diagnosis, the performance at a population level is probably far less important to you than how reliably and consistently the test will assign you to the right treatment group.

In practice, this approach is relatively simple (Haibe-Kains et al. 2012; Beck et al. 2013). One starts with multiple independent training sets for which the samples have a known classification and a single independent test set.

- 1) For each training set, one identifies a gene set that distinguishes the classes; alternatively, one can start with a candidate gene set based on some other criteria (such as a representative gene set from known pathways or culled from the literature).
- 2) For each training set, and for the appropriate gene set, one fits the classification model.
- 3) Each classification model is used independently to predict the classes of the samples in the test set.
- 4) The concordance between different classifiers is used to estimate the robustness of the method, measuring how often classifiers produced with different training sets give the same classification for each sample.

In many ways, this robustness assessment speaks to our intuition about how a good classification model should perform. It should not matter if the model was learned in London, Paris, Sydney, Tokyo, or Boston. If the gene set and the predictive model are truly reliable, they should give the same classification.

In practice, this approach can lead to the same difficulties that arise in cross-validation – namely, that there is no single classifier. However, given that a good classifier is going to be highly concordant independent of which dataset was used to train the method, one would hope that all are more or less interchangeable. In any event, this means of evaluation should be considered complementary to, but not a replacement for, validation using an independent test set.

Single-Cell Sequencing

Although single-cell RNA sequencing (scRNA-seq) is relatively new, it is particularly exciting as it allows the transcriptomes of individual cells to be profiled. Although the first scRNA-seq experiments analyzed a single cell or a very small number of cells, we can now analyze thousands of cells per experiment and there are a host of published protocols (SMART-Seq [Ramskold et al. 2012], CEL-Seq [Hashimshony et al. 2012], and Drop-Seq [Macosko et al. 2015]) in use as well as robust commercial products such as the 10x Chromium system.

Unlike conventional applications of DNA microarrays and RNA-seq that analyze gene expression in bulk tissue samples, scRNA-seq uses barcoding to create sequencing libraries from each cell in a sample, producing expression profiles for individual cells in the original sample. As such, scRNA-seq not only allows expression to be compared between phenotypes, but also to define cell populations and to study how expression variation and cellular heterogeneity is associated with phenotype. Published applications of scRNA-seq include identification and exploration of cell types and their diversity, analyses of the

stochasticity of gene expression, and estimation of gene regulatory networks across cells and cell types.

The processing steps used in scRNA-seq are similar to those used in RNA-seq. Typically, one performs quality control on sequence reads, aligning quality reads to an appropriate reference (using, for example, Salmon [Patro et al. 2017] or Kallisto [Bray et al. 2016]), and mapping quality control. At this stage, one typically uses methods that are unique to scRNA-seq for normalization, for identifying subgroups based on expression, for testing for differential expression, and for doing functional analysis. These include SCONE (Cole et al. 2018) for normalization, Seurat (Butler et al. 2018) and GiniClust/GiniClust2 (Jiang et al. 2016; Tsoucas and Yuan 2018) for finding cell populations, and web-based processing platforms such as Falco (Yang et al. 2017) and ASAP (Chen et al. 2017; Gardeux et al. 2017) for providing integrated analyses.

The reason the analysis path of scRNA-seq diverges from the analysis of bulk RNA-seq is that there are a number of technical artifacts that are specific to scRNA-seq. One example is the batch effects that occur as a result of cell isolation and amplification. But a more significant problem is that of gene “drop-out” or sparsity. It is estimated that each cell has as many as 300 000 RNA transcripts at any one time. However, in scRNA-seq, only a few thousands (or tens of thousands) of reads per cell are typically recorded. Drop-out occurs owing to statistics associated with the mRNA counting experiment – we miss some transcripts by chance. While highly expressed transcripts are generally well represented in scRNA-seq data, moderately to lowly expressed transcripts can be missed. And the overall pattern of “drop-out” genes produces a cells-by-transcripts expression matrix that is sparse – meaning that there are far fewer observed counts that might be expected. Fortunately, this problem has been explored in the microbiome literature and many methods in use for scRNA-seq can trace their roots to microbiome analysis.

However, these issues mean that, as of the time of writing, there is no general consensus on many aspects of scRNA-seq analysis. For example, in normalization, both global and scaling methods are frequently used and the TPM and CPM measures are common. In comparing expression, methods are being developed that account for the multi-modularity of gene expression that results from cell populations. However, many published studies still rely on established methods for phenotype comparison, including the *t*-test.

A new type of visualization that is now widely used in the scRNA-seq literature is *t*-distributed stochastic neighbor embedding (*t*-SNE) plots (van der Maaten and Hinton 2008). Similar to methods such as PCA, *t*-SNE reduces high dimensional data, but does so by solving an objective function that attempts to preserve distances between similar genes or cells. While *t*-SNE has proven to be extremely useful, it is memory and computationally intensive and the run time scales with the square of the number of cells or genes analyzed.

Summary

This chapter provides a roadmap for the analysis of gene expression data and is, by the very nature of the rapid changes in this field, incomplete. Since the first genome-wide analysis of expression through the sequencing of expressed sequence tags was performed in the early 1990s, changes in technology, advances in analytical methods, and an explosion of ancillary data (such as genome sequences from many organisms, as well as their gene annotations) have transformed the field. While the choice of particular software tools or analytical methods can be debated, and the best options today will be eclipsed by new methods tomorrow, the general principles of good experimental design and sound analytical practice remain unchanged. Rather than a cookbook, this chapter is best considered as a roadmap, guiding the researcher along a path that will increase the likelihood of success and provide some confidence in the results. In that spirit, we hope you find the methods outlined here a useful introduction and guide.

Internet Resources

ArrayExpress	www.ebi.ac.uk/arrayexpress
Bioconductor	www.bioconductor.org
Database for Annotation, Visualization and Integrated Discovery (DAVID)	david.ncifcrf.gov
Gene Expression Omnibus (GEO)	www.ncbi.nlm.nih.gov/geo
Gene Set Enrichment Analysis (GSEA)	software.broadinstitute.org/gsea/index.jsp
Genomic Data Commons (GDC) Data Portal	portal.gdc.cancer.gov
Genotype Tissue-Expression project (GTEx)	gtexportal.org

Further Reading

- Brazma, A., Hingamp, P., Quackenbush, J. et al. (2001). Minimum information about a microarray experiment (MIAME) – toward standards for microarray data. *Nat. Genet.* 29 (4): 365–371. <https://doi.org/10.1038/ng1201-365>. This foundational paper established the principles for public access of well-annotated genomic datasets and has stood the test of time. The reporting criteria laid out in the MIAME standards should be considered when designing and reporting all large-scale experiments.
- Conesa, A., Madrigal, P., Tarazona, S. et al. (2016). A survey of best practices for RNA-seq data analysis. *Genome Biol.* 17: 13. <https://doi.org/10.1186/s13059-016-0881-8>.
- Conesa, A., Madrigal, P. et al. (2016). Erratum to: A survey of best practices for RNA-seq data analysis. *Genome Biol.* 17 (1): 181. <https://doi.org/10.1186/s13059-016-1047-4>. This paper (and the erratum) provides a nice survey of RNA-seq data analysis methods as well as guidance to best practices. Although the field is rapid evolving, the “best” practices are largely extensions of the hard lessons learned from DNA microarray analysis.
- Ching, T., Huang, S., and Garmire, L.X. (2014). Power analysis and sample size estimation for RNA-seq differential expression. *RNA.* 20 (11): 1684–1696. <https://doi.org/10.1261/rna.046011.114>. Power calculations for gene expression analysis are notoriously difficult, in large part because expression levels and variance differ substantially between genes. This paper provides some guidance on methods to estimate power and the sample size necessary for RNA-seq studies.
- Paulson, J.N., Chen, C.Y., Lopes-Ramos, C.M. et al. (2017). Tissue-aware RNA-seq processing and normalization for heterogeneous and sparse data. *BMC Bioinf.* 18 (1): 437. <https://doi.org/10.1186/s12859-017-1847-x>. This paper describes a simple data quality control and normalization pipeline that can be used to check (some) sample annotation and normalize data from heterogeneous samples.
- Glass, K., Huttenhower, C., Quackenbush, J., and Yuan, G.C. (2013). Passing messages between biological networks to refine predicted interactions. *PLoS One.* 8 (5): e64832. <https://doi.org/10.1371/journal.pone.0064832>. There are many methods for inferring gene regulatory networks. We developed this method because it builds on our understanding that transcription factors regulate expression and on the assumption that networks differ between phenotypes.
- Hung, J.H., Yang, T.H., Hu, Z. et al. (2012). Gene set enrichment analysis: performance evaluation and usage guidelines. *Briefings Bioinf.* 13 (3): 281–291. <https://doi.org/10.1093/bib/bbr049>. Although gene set enrichment methods continue to evolve, this review does a good job of comparing various methods, highlighting strengths and weaknesses, and giving insight into best practices.

References

- Alexa, A., Rahnenfuhrer, J., and Lengauer, T. (2006). Improved scoring of functional groups from gene expression data by decorrelating GO graph structure. *Bioinformatics*. 22 (13): 1600–1607. <https://doi.org/10.1093/bioinformatics/btl140>.
- Anders, S. and Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biol*. 11 (10): R106. <https://doi.org/10.1186/gb-2010-11-10-r106>.
- Beck, A.H., Knoblauch, N.W., Hefti, M.M. et al. (2013). Significance analysis of prognostic signatures. *PLoS Comput. Biol*. 9 (1): e1002875. <https://doi.org/10.1371/journal.pcbi.1002875>.
- Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. Series B Methodol*. 57 (1): 289–300.
- Benjamini, Y. and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *Ann. Statist*. 29 (4): 1165–1188. <https://doi.org/10.1214/aos/1013699998>.
- Bolstad, B.M., Irizarry, R.A., Astrand, M., and Speed, T.P. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*. 19 <https://doi.org/10.1093/bioinformatics/19.2.185>.
- Bolstad, B.M., Collin, F., Simpson, K.M. et al. (2004). Experimental design and low-level analysis of microarray data. *Int. Rev. Neurobiol*. 60: 25–58.
- Bray, N.L., Pimentel, H., Melsted, P., and Pachter, L. (2016). Near-optimal probabilistic RNA-seq quantification. *Nat. Biotechnol*. 34 (5): 525–527. <https://doi.org/10.1038/nbt.3519>.
- Brettschneider, J., Collin, F., Bolstad, B.M., and Speed, T.P. (2008). Quality assessment for short oligonucleotide microarray data. *Technometrics*. 50 (3): 241–264.
- Butler, A., Hoffman, P., Smibert, P. et al. (2018). Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat. Biotechnol*. 36 (5): 411–420. <https://doi.org/10.1038/nbt.4096>.
- Callow, M.J., Dudoit, S., Gong, E.L. et al. (2000). Microarray expression profiling identifies genes with altered expression in HDL-deficient mice. *Genome Res*. 10 (12): 2022–2029.
- Chen, W., Gardeux, V., Meireles-Filho, A., and Deplancke, B. (2017). Profiling of single-cell transcriptomes. *Curr. Protoc. Mouse Biol*. 7 (3): 145–175. <https://doi.org/10.1002/cpmo.30>.
- Cole, M.B., Risso, D., Wagner, A. et al. (2018). Performance assessment and selection of normalization procedures for single-cell RNA-seq. *bioRxiv* [biorxiv.org/content/early/2018/05/18/235382.abstract](https://doi.org/10.1101/235382).
- De Jay, N., Papillon-Cavanagh, S., Olsen, C. et al. (2013). mRMRe: an R package for parallelized mRMR ensemble feature selection. *Bioinformatics*. 29 (18): 2365–2368. <https://doi.org/10.1093/bioinformatics/btt383>.
- DeRisi, J., Penland, L., Brown, P.O. et al. (1996). Use of a cDNA microarray to analyse gene expression patterns in human cancer. *Nat. Genet*. 14 (4): 457–460.
- Ding, C. and Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *J. Bioinform. Comput. Biol*. 3 (2): 185–205.
- Dobin, A., Davis, C.A., Schlesinger, F. et al. (2013). STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* 29 (1): 15–21. <https://doi.org/10.1093/bioinformatics/bts635>.
- Eisen, M.B., Spellman, P.T., Brown, P.O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*. 95 (25): 14863–14868.
- Gardeux, V., David, F.P.A., Shajkofci, A. et al. (2017). ASAP: a web-based platform for the analysis and interactive visualization of single-cell RNA-seq data. *Bioinformatics*. 33 (19): 3123–3125. <https://doi.org/10.1093/bioinformatics/btx337>.
- Gautier, L., Cope, L., Bolstad, B.M., and Irizarry, R.A. (2004). affy—analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics*. 20 (3): 307–315.
- Golub, T.R., Slonim, D.K., Tamayo, P. et al. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*. 286 (5439): 531–537.

- Haibe-Kains, B., Desmedt, C., Loi, S. et al. (2012). A three-gene model to robustly identify breast cancer molecular subtypes. *J. Natl. Cancer Inst.* 104 (4): 311–325. <https://doi.org/10.1093/jnci/djr545>.
- Hashimshony, T., Wagner, F., Sher, N., and Yanai, I. (2012). CEL-Seq: single-cell RNA-seq by multiplexed linear amplification. *Cell Rep.* 2 (3): 666–673. <https://doi.org/10.1016/j.celrep.2012.08.003>.
- Hastie, T., Tibshirani, R., and Friedman, J.H. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Predictions*. New York, NY: Springer.
- Hastie, T., Tibshirani, R., and Friedman, J.H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2e. New York, NY: Springer.
- Hegde, P., Qi, R., Abernathy, K. et al. (2000). A concise guide to cDNA microarray analysis. *Biotechniques* 29 (3): 548–550, 52–44, 56, passim.
- da Huang, W., Sherman, B.T., and Lempicki, R.A. (2009a). Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Res.* 37 (1): 1–13. <https://doi.org/10.1093/nar/gkn923>.
- da Huang, W., Sherman, B.T., and Lempicki, R.A. (2009b). Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nat. Protoc.* 4 (1): 44–57. <https://doi.org/10.1038/nprot.2008.211>.
- Ioannidis, J.P., Allison, D.B., Ball, C.A. et al. (2009). Repeatability of published microarray gene expression analyses. *Nat. Genet.* 41 (2): 149–155.
- Irizarry, R.A., Bolstad, B.M., Collin, F. et al. (2003). Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Res.* 31 (4): e15.
- Irizarry, R.A., Warren, D., Spencer, F. et al. (2005). Multiple-laboratory comparison of microarray platforms. *Nat. Methods* 2 (5): 345–350. <https://doi.org/10.1038/nmeth756>.
- Ishmael, N., Dunning Hotopp, J.C., Ioannidis, P. et al. (2009). Extensive genomic diversity of closely related *Wolbachia* strains. *Microbiology* 155 (Pt 7): 2211–2222.
- Jiang, L., Chen, H., Pinello, L., and Yuan, G.C. (2016). GiniClust: detecting rare cell types from single-cell gene expression data with Gini index. *Genome Biol.* 17 (1): 144. <https://doi.org/10.1186/s13059-016-1010-4>.
- Johnson, W.,E., Li, C., and Rabinovic, A. (2007). Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics.* 8 (1): 118–127. <https://doi.org/10.1093/biostatistics/kxj037>.
- Kahvejian, A., Quackenbush, J., and Thompson, J.F. (2008). What would you do if you could sequence everything? *Nat. Biotechnol.* 26 (10): 1125–1133. <https://doi.org/10.1038/nbt1494>.
- Konstantinopoulos, P.A., Cannistra, S.A., Fountzilias, H. et al. (2011). Integrated analysis of multiple microarray datasets identifies a reproducible survival predictor in ovarian cancer. *PLoS One* 6 (3): e18202.
- Lander, E.S., Linton, L.M., Birren, B. et al., International Human Genome Sequencing Consortium (2001). Initial sequencing and analysis of the human genome. *Nature* 409 (6822): 860–921. <https://doi.org/10.1038/35057062>.
- Langmead, B. and Salzberg, S.L. (2012). Fast gapped-read alignment with Bowtie 2. *Nat. Methods* 9 (4): 357–359. <https://doi.org/10.1038/nmeth.1923>.
- Langmead, B., Trapnell, C., Pop, M., and Salzberg, S.L. (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.* 10 (3): R25. <https://doi.org/10.1186/gb-2009-10-3-r25>.
- Larkin, J.E., Frank, B.C., Gavras, H. et al. (2005). Independence and reproducibility across microarray platforms. *Nat. Methods.* 2 (5): 337–344. <https://doi.org/10.1038/nmeth757>.
- Leek, J.T., Johnson, W.E., Parker, H.S. et al. (2012). The SVA package for removing batch effects and other unwanted variation in high-throughput experiments. *Bioinformatics.* 28 (6): 882–883. <https://doi.org/10.1093/bioinformatics/bts034>.
- Li, H. and Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics.* 25 (14): 1754–1760. <https://doi.org/10.1093/bioinformatics/btp324>.

- Li, P., Piao, Y., Shon, H.S., and Ryu, K.H. (2015). Comparing the normalization methods for the differential analysis of Illumina high-throughput RNA-seq data. *BMC Bioinf.* 16: 347. <https://doi.org/10.1186/s12859-015-0778-7>.
- Love, M.I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.* 15 (12): 550. <https://doi.org/10.1186/s13059-014-0550-8>.
- van der Maaten, L. and Hinton, G.E. (2008). Visualizing high-dimensional data using t-SNE. *J. Machine Learn. Res.* 9: 2579–2605. prlab.tudelft.nl/sites/default/files/vandermaaten08a.pdf.
- Macosko, E.Z., Basu, A., Satija, R. et al. (2015). Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell.* 161 (5): 1202–1204. <https://doi.org/10.1016/j.cell.2015.05.002>.
- Michaels, G.S., Carr, D.B., Askenazi, M. et al. (1998). Cluster analysis and data visualization of large-scale gene expression data. *Pac. Symp. Biocomput* 1998: 42–53.
- Nagalakshmi, U., Wang, Z., Waern, K. et al. (2008). The transcriptional landscape of the yeast genome defined by RNA sequencing. *Science* 320 (5881): 1344–1349. <https://doi.org/10.1126/science.1158441>.
- Oron, A.P., Jiang, Z., and Gentleman, R. (2008). Gene set enrichment analysis using linear models and diagnostics. *Bioinformatics.* 24 (22): 2586–2591. <https://doi.org/10.1093/bioinformatics/btn465>.
- Patro, R., Mount, S.M., and Kingsford, C. (2014). Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms. *Nat. Biotechnol.* 32 (5): 462–464. <https://doi.org/10.1038/nbt.2862>.
- Patro, R., Duggal, G., Love, M.I. et al. (2017). Salmon provides fast and bias-aware quantification of transcript expression. *Nat. Methods.* 14 (4): 417–419. <https://doi.org/10.1038/nmeth.4197>.
- Paulson, J.N., Chen, C.Y., Lopes-Ramos, C.M. et al. (2017). Tissue-aware RNA-seq processing and normalization for heterogeneous and sparse data. *BMC Bioinf.* 18 (1): 437. <https://doi.org/10.1186/s12859-017-1847-x>.
- Perou, C.M., Jeffrey, S.S., van de Rijn, M. et al. (1999). Distinctive gene expression patterns in human mammary epithelial cells and breast cancers. *Proc. Natl. Acad. Sci. USA.* 96 (16): 9212–9217.
- Pop, M., Paulson, J.N., Chakraborty, S. et al. (2016). Individual-specific changes in the human gut microbiota after challenge with enterotoxigenic *Escherichia coli* and subsequent ciprofloxacin treatment. *BMC Genomics.* 17: 440. <https://doi.org/10.1186/s12864-016-2777-0>.
- Quackenbush, J. (2005). Extracting meaning from functional genomics experiments. *Toxicol. Appl. Pharmacol.* 207 (2 Suppl): 195–199.
- Ramskold, D., Luo, S., Wang, Y.C. et al. (2012). Full-length mRNA-seq from single-cell levels of RNA and individual circulating tumor cells. *Nat. Biotechnol.* 30 (8): 777–782. <https://doi.org/10.1038/nbt.2282>.
- Robinson, M.D., McCarthy, D.J., and Smyth, G.K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* 26 (1): 139–140. <https://doi.org/10.1093/bioinformatics/btp616>.
- Schena, M., Shalon, D., Davis, R.W., and Brown, P.O. (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science.* 270 (5235): 467–470.
- Simon, R., Radmacher, M.D., and Dobbin, K. (2002). Design of studies using DNA microarrays. *Genet. Epidemiol.* 23 (1): 21–36.
- Spellman, P.T., Sherlock, G., Zhang, M.Q. et al. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell* 9 (12): 3273–3297.
- Subramanian, A., Tamayo, P., Mootha, V.K. et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. USA.* 102 (43): 15545–15550. <https://doi.org/10.1073/pnas.0506580102>.

- Toker, L., Feng, M., and Pavlidis, P. (2016). Whose sample is it anyway? Widespread misannotation of samples in transcriptomics studies. *F1000Res*. 5: 2103. <https://doi.org/10.12688/f1000research.9471.2>.
- Tsoucas, D. and Yuan, G.C. (2018). GiniClust2: a cluster-aware, weighted ensemble clustering method for cell-type detection. *Genome Biol*. 19 (1): 58. <https://doi.org/10.1186/s13059-018-1431-3>.
- Venet, D., Dumont, J.E., and Detours, V. (2011). Most random gene expression signatures are significantly associated with breast cancer outcome. *PLoS Comput. Biol*. 7 (10): e1002240. <https://doi.org/10.1371/journal.pcbi.1002240>.
- Venter, J.C., Adams, M.D., Myers, E.W. et al. (2001). The sequence of the human genome. *Science*. 291 (5507): 1304–1351.
- Wen, X., Fuhrman, S., Michaels, G.S. et al. (1998). Large-scale temporal gene expression mapping of central nervous system development. *Proc. Natl. Acad. Sci. USA*. 95 (1): 334–339.
- Wilson, C.L. and Miller, C.J. (2005). Simpleaffy: a BioConductor package for Affymetrix Quality Control and data analysis. *Bioinformatics*. 21 (18): 3683–3685.
- Yang, A., Troup, M., Lin, P., and Ho, J.W. (2017). Falco: a quick and flexible single-cell RNA-seq processing framework on the cloud. *Bioinformatics*. 33 (5): 767–769. <https://doi.org/10.1093/bioinformatics/btw732>.

11

Proteomics and Protein Identification by Mass Spectrometry

Sadhna Phanse and Andrew Emili

Introduction

What Is a Proteome?

A proteome is the entire set of proteins expressed in a biological entity (cell, tissue, organ, or organism) at any point in time during its life cycle. It is derived from the words *protein* and *genome* and was first coined by Marc Wilkins in 1995 in reference to the functional study of proteins using mass spectrometry (MS) (Wilkins et al. 1996). Proteomics is the large-scale study of proteins that employs a systematic, shotgun, or targeted high-throughput approach to elucidate their identity, localization, abundance, structure, function, or expression profiles. Proteomics complements other “omics” studies, such as genomics or transcriptomics, to expand on the identity of proteins that are encoded by genes and to determine their fundamental role in the cell. While a genome of an organism is relatively static, the proteome is highly dynamic and differs from cell to cell and changes – in terms of the abundance, post-translational modifications (PTMs), and stability and physical associations of the expressed protein isoforms – in response to different environmental stimuli. This dynamic, ever-changing nature makes the proteome significantly more complex than the genome. For instance, the human genome comprises ~20 000 protein-coding open reading frames (Gaudet et al. 2017). On the other hand, mutations and alternative transcription and splicing isoforms and other mechanisms can give rise to multiple different messenger RNA (mRNA) transcripts from a single gene (Figure 11.1). In addition, site-specific chemical or enzymatic modifications during or after translation can result in various diverse proteoforms (i.e. different forms of proteins) that change over time, subcellular location, and physiological or diseased conditions. This suggests that the human proteome may actually consist of millions of chemically distinct entities.

Why Study Proteomes?

Major advances in DNA sequencing over the last decade have resulted in the determination of the complete genomes of over 8000 organisms and the availability of partial draft genomes for approximately another 37 000 species (Mukherjee et al. 2017). As a consequence, there has been an exponential increase in the number of putative protein sequences, or “virtual proteomes,” which has, in turn, created a vital need for the determination of the physical, structural, and functional roles played by these proteins. Given the complex, dynamic nature of the proteome, it is important to not only identify the cognate gene from which an expressed protein comes but also to identify what form and what associations the corresponding protein takes under particular biological circumstances. This is often called *protein characterization*. In the 1990s, the advent of biological MS as a flexible, sensitive, and rapid way to identify and quantify proteins in complex biological mixtures helped usher in the proteomics era

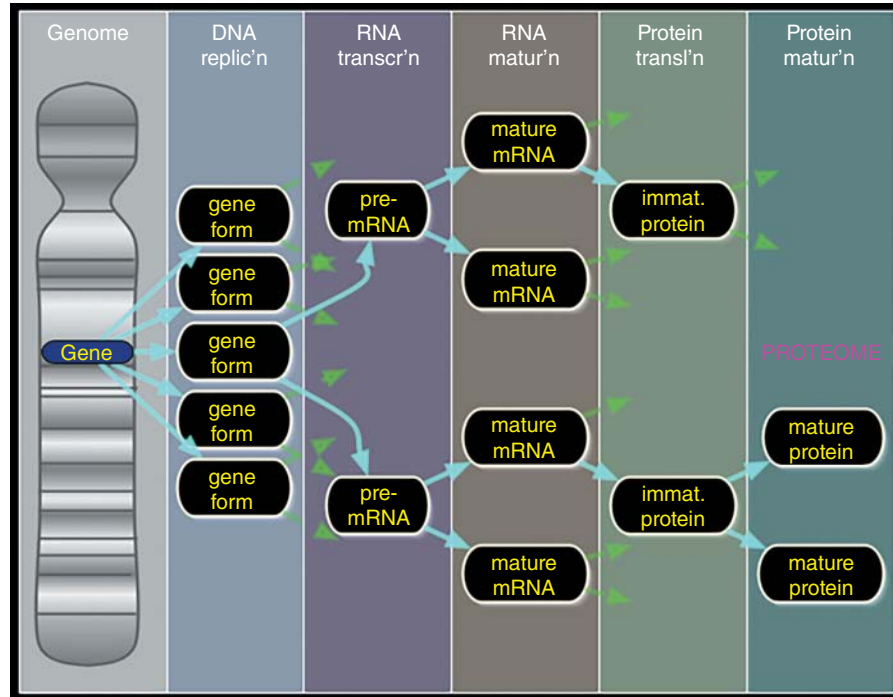


Figure 11.1 Gene(s) to proteoforms. This figure illustrates the complexity of the proteome. While the one-gene, one-protein paradigm still generally holds true, as demonstrated by the Gene → DNA replication → RNA transcription → Protein translation → Protein model, it is complicated by the fact that there exist alternative means to create protein variation at each step. These include variants arising due to DNA polymorphisms, RNA polymerase slippage, alternative splicing, and RNA editing; protein translation frameshifts generating recoded fusion proteins; as well as protein cleavage and diverse post-translational modifications. All of these can result in different proteoforms arising from the same gene.

(Pandey and Mann 2000). In turn, the publication of a draft human genome, the availability of annotated protein sequences in public databases, the introduction of high-performance MS platforms, and the development of better computational tools made the advent of high-throughput proteome-scale analysis of proteins possible.

Before the application of MS to proteomics, the analysis of proteins was primarily done using gel separation techniques such as two-dimensional gel electrophoresis (2DGE), followed by traditional identification with Edman protein sequencing or the newer protein antibody arrays. Although 2DGE was considered effective in terms of soluble protein separation for hydrophilic proteins, the ability to identify the separated proteins was limited and time-consuming. However, breakthrough advances in protein MS have now made the large-scale study of proteins possible. In particular, the development of electrospray ionization (ESI) and matrix-assisted laser desorption ionization (MALDI) techniques by John Fenn and Koichi Tanaka, who shared the 2002 Nobel Prize in Chemistry, were key. Likewise, advances in sample preparation protocols to handle small biological specimens, the development of database search algorithms to identify known proteins using sequence databases (Eng et al. 1994), and the direct analysis of increasingly complex protein mixtures (Aebersold and Mann 2003) have steadily increased the power and utility of MS-based proteomics. Owing to these developments, it is now possible to routinely identify and quantify thousands of proteins (and their modifications) with higher speed and greater precision than ever before.

This chapter examines some of the more popular MS techniques, their affiliated software tools, and database resources used in the interpretation and analysis of proteomics data. The primary focus will be on protein identification, PTM mapping, and expression profiling through the use of bioinformatics.

Mass Spectrometry

MS is a versatile analytical technique that can precisely measure the molecular mass of the compounds present in a sample. Molecular mass (or molecular weight) is a fundamental property of all elements, chemicals, and molecules. If it is very accurately measured, one can determine the molecular formula and even the structure of a given chemical compound. The basic principle of MS is to generate charged gas-phase ions from organic or inorganic compounds in a specimen, followed by the separation and detection of these ions based on their mass-to-charge ratio (m/z) and intensity (abundance). A mass spectrometer is commonly made up of a sample ionizer, a mass analyzer, and a detector. The ionizer forms the gaseous ions from the specimen of interest, for instance, by shooting the specimen with a laser. The mass of the ions is determined using the mass analyzer, which separates the ions based on their m/z ratios and directs the different ions to the detector where they are sensed electronically; the corresponding signal is converted to a digital output in the form of a mass spectrum.

Ionization

While MS has been the standard tool for the analysis of volatile organic compounds since the 1950s, its use in the field of proteomics gained momentum with the development of soft ionization techniques like ESI (Fenn et al. 1989) and MALDI (Karas and Hillenkamp 1988) in the 1980s. In ESI, the liquid sample is sprayed through a needle capillary into an ionization source. A high voltage (in either positive or negative mode) is applied between the outlet of the sample stream and the inlet of the mass analyzer. The liquid continuously absorbs the charge, becomes unstable, and is released in the form of tiny highly charged droplets. Evaporation of solvent from the spray, which can be facilitated by passage through a stream of drying gas (e.g. nitrogen), generates charged analyte ions. Conversely, MALDI uses an ultraviolet laser beam in a vacuum to blast (and ionize) sample molecules embedded in a chemical matrix situated on a target plate. The matrix is typically made with highly conjugated organic acids such as the commonly used 2,5-dihydroxybenzoic acid (DHB). The matrix sublimates into a gaseous cloud by absorbing energy in the form of heat, causing desorption while allowing the analyte to remain intact. The collision of molecules in the gaseous cloud causes the transfer of energy from the matrix to the analyte. De-solvation occurs by proton transfer between the excited matrix and analyte, resulting in protonated/deprotonated ions. The gentle but effective ESI and MALDI methods, described above, that allow protein or peptide molecules to remain relatively intact during the ionization process markedly raised the dynamic upper mass range of detection from <1000 Da to >500 000 Da, thus increasing the efficiency of polypeptide detection by MS and making the routine analysis of the protein components of biological samples possible.

Mass Analyzers

Ions produced by any of the aforementioned ionization methods can be sorted and measured by mass analyzers. There are multiple types of mass analyzers associated with routine protein analysis, each differing in the fundamental way in which they separate or fragment ions, in their accuracy in determining the ion mass (mass precision), in their ability to distinguish components with the same nominal or unit mass (resolution), and with respect to the range of m/z ratios that can be measured by the mass analyzer (dynamic mass range capability). Quadrupole, time of flight (TOF), Fourier transform ion cyclotron resonance (FT-ICR), ion trap, and Orbitrap all represent major categories of mass analyzers, although there are numerous variations available within each class. The quadrupole mass analyzer, which is a low-resolution analyzer, is made up of four charged rods set in a grid and uses alternating quadrupolar electric fields for the rapid separation and selection (transmission) of ions of interest (Figure 11.2). By controlling the applied voltage, ions with certain m/z ratios can be

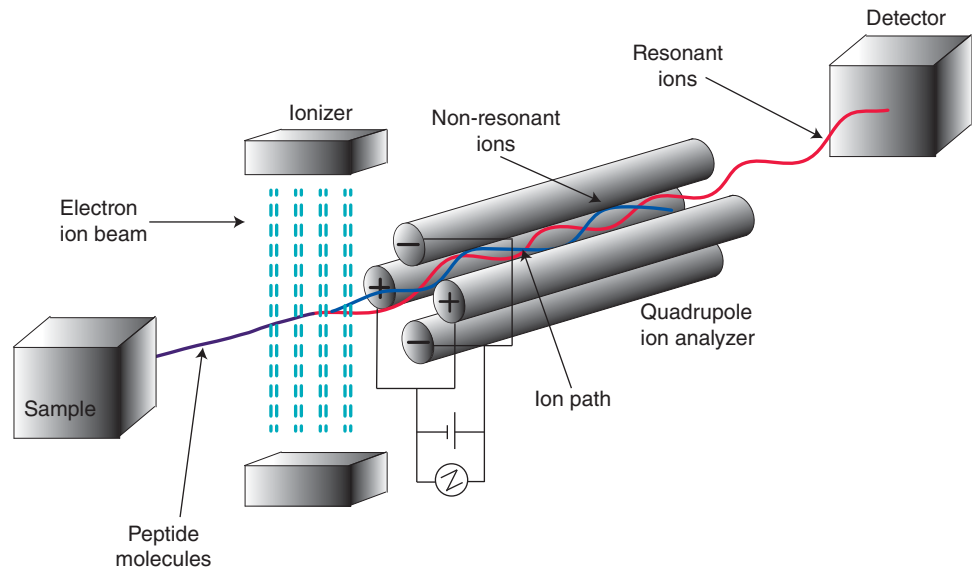


Figure 11.2 Quadrupole mass analyzer. Schematic of a quadrupole mass analyzer, made up of four parallel cylindrical rods with each opposing pair connected electrically and a radio frequency voltage with DC offset applied between them. Ions travel down the quadrupole between the rods; only molecules with a certain m/z ratio having a stable trajectory (resonant ions) in the oscillating electrical field are allowed to reach the detector for a given voltage offset, allowing filtering of sample ions. Ions with a weak trajectory (non-resonant ions) strike the rods and are lost.

qualitatively selected and transferred to the detector. In a TOF mass analyzer, ions are accelerated by an electric field of known strength. As the initial velocity of the ions depends on their m/z ratio, they reach the detector at different times, with lighter and/or more charged ions reaching first. Hence, ions can be distinguished by their “TOF” to the analyzer (Figure 11.3). Ion trap analyzers use a combination of magnetic and electric fields to capture ions in an isolated environment. Ions can be trapped using the Penning trap (FT-ICR), the Paul ion trap (quadrupole ion trap), the Kingdon trap, and the Orbitrap – a radically improved implementation of the Kingdon trap. Various combinations of these mass analyzers are in wide use.

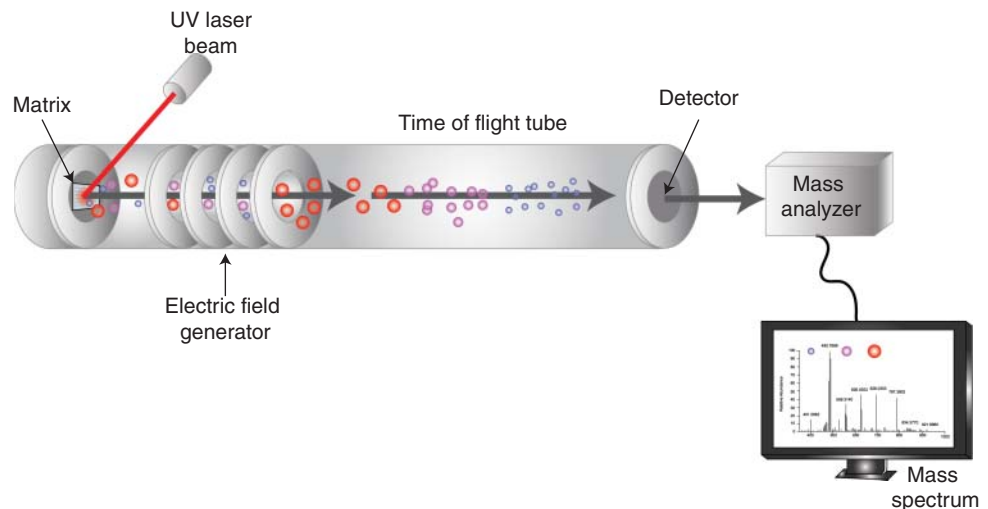


Figure 11.3 Time of flight (TOF) mass analyzer. Schematic of a TOF mass analyzer, where the mass-to-charge ratio is determined by the time taken by the ions to reach the detector. Ions are accelerated by applying an electrical field of known voltage and passing through a time-of-flight tube. The velocity of each ion is based on its mass-to-charge ratio, so ions with lower m/z reach the detector before those with higher m/z .

A triple quadrupole mass analyzer is a variation of the quadrupole analyzer that uses a linear series of three quadrupoles (essentially two mass spectrometers connected together by a central quadrupole) for improved sensitivity and resolution. The central quadrupole can be used to fragment ions, allowing one to perform a very useful technique called tandem MS, popularly referred to as MS/MS or MS², where selected ions of interest detected after passage through the first analyzer undergo fragmentation in the second unit before detection in the third (Box 11.1). Hence, in certain configurations, the mass spectrometer can fragment selected ion species in order to deduce their corresponding molecular structures (e.g. polypeptide sequences) or to derive deeper structural information (e.g. protein PTMs or folding states).

Box 11.1 Tandem Mass Spectrometry (Figure 11.4)

- *Tandem mass spectrometry (MS)* is an MS technique that involves multiple rounds of analysis. Typically, ions formed in the ion source are separated according to their m/z ratio in the first round of mass analysis (MS¹). Ions are selected sequentially based on their m/z ratio (precursor ions) and relative intensity, then subjected to fragmentation through molecular activation by increasing their internal energy. The resulting product ions are, in turn, separated and detected in the second stage of mass analysis (MS²). Fragmentation results from the dissociation of molecular ions formed in the first round of analysis and is a critical component of tandem MS. The activation methods used to fragment ions can be collisional, electron based, or involve photo-activation. Popular ion activation/fragmentation methods are collision-induced dissociation (CID; Jennings 1968), electron capture dissociation (ECD; Zubarev et al. 1998), electron transfer dissociation (ETD; Syka et al. 2004), higher energy collisional activation dissociation (HCD; Olsen et al. 2007), and infrared multi-photon dissociation (IRMPD; Little et al. 1994).
- *Collisional fragmentation.* CID, also known as collisional-activated dissociation (CAD), is a commonly used technique to fragment ions in the gas phase. CID involves energetic collisions between ions of interest and non-reactive gas atoms (typically helium, nitrogen, or argon). During collisions, the kinetic energy of a molecular ion is converted to internal energy, the accumulation of which results in bond breakage and dissociation of the precursor ions into smaller fragments for subsequent detection by MS². HCD is a CID technique specific to Orbitrap, in which fragmentation takes place external to the trap. The high efficiency of most collisional methods makes them a top choice in virtually all MS² proteomics studies.
- *Photo-activated fragmentation.* IRMPD is a method that uses an infrared laser beam to increase the internal energy of the trapped ions. The photons in the laser beam are absorbed by the trapped ions, creating a vibrationally excited state that leads to release of energy via bond dissociation, very similar to CID. The ions commonly generated through collisional or photo-activated fragmentation are the *b* and *y* ions formed by the dissociation of the weak amide bonds. These techniques are quite efficient with regards to analysis relating to peptides, lipids, and small molecules but may remove PTMs.
- *Electron-based fragmentation.* In ECD, peptide ions of interest are irradiated with low-energy electrons (~0.2 eV), resulting in the capture of an electron and producing an unstable charge-reduced species, which dissociates to give fragment ions that are informative about peptide sequence. ETD is analogous to ECD, but dissociation is induced by the transfer of an electron between oppositely charged ions. Regardless, in both ECD and ETD, fragmentation occurs because of the cleavage of the N–C α bonds, giving rise to complementary *c* and *z* ions. ECD and ETD are now widely applied to the study of full-length proteins (so-called “top-down” sequencing) and peptides with labile PTMs, such as phosphorylation.

(Continued)

Box 11.1 (Continued)

One disadvantage of MS/MS methods that use vibrational excitation (such as CID for peptide fragmentation) is that they can cause biased cleavage of certain weaker bonds present in either the peptide backbone or side chains. These include PTMs, such as phosphate side groups, as a preferred site of cleavage, resulting in loss of PTM sites and reduced complexity spectra that are hard to interpret on the sequence level. This, in turn, leads to missed or incorrect identifications and site assignments. In contrast, ETD is a gentler method of fragmentation that makes use of the low-energy electron transfer through a more comprehensive non-ergodic process that preserves the modification sites of PTMs, making it a method of choice for the fragmentation of PTMs.

Relative to quadrupole or triple quadrupole analyzers, TOF mass analyzers offer a much higher mass resolution for analyzing polypeptide ions and their fragments, while FT-ICR and Orbitrap mass analyzers offer the highest mass resolution of all analyzers but have more limited dynamic range.

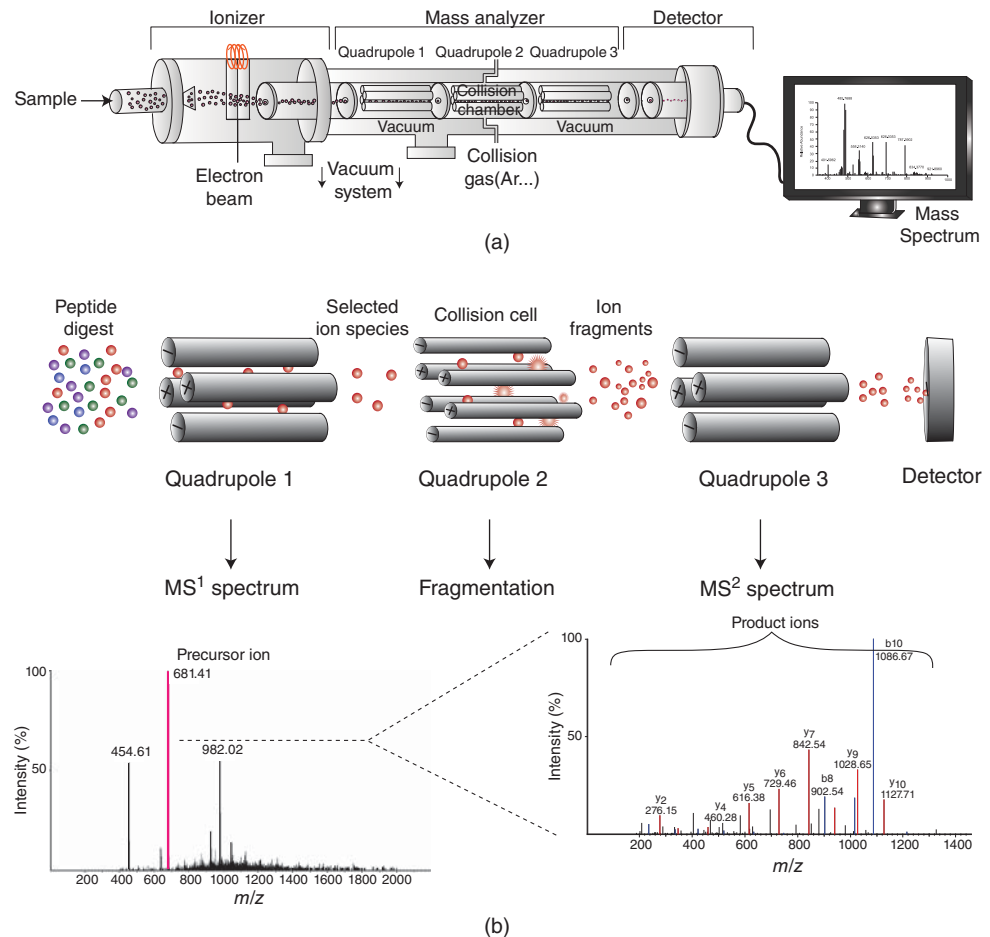


Figure 11.4 (a) Tandem mass spectrometry (MS). Schematic of a triple quadrupole mass spectrometer used in tandem MS for peptide sequencing. (b) The first stage of liquid chromatography tandem MS analysis is carried out as an MS¹ precursor ion scan (quadrupole 1). In the second stage, the instrument is operated in the MS² mode wherein a selected precursor ion (defined as m/z) is passed to a collision chamber (quadrupole 2) for fragmentation (e.g. by interactions with inert gas). The resulting peptide ion fragments are then resolved on the basis of their apparent m/z ratios in quadrupole 3.

Ion Detectors

After passing through the analyzer, the separated (peptide) ions strike the detector (or ion collection system) and are then identified according to their m/z ratio and relative intensity (which is correlated with abundance). Detectors are capable of signal amplification, and some are sensitive enough to potentially pick out single molecules. There are several types of detectors available for mass spectrometers. The most routinely used detector is the electron multiplier, which operates by detecting the secondary electron emission produced by the charged ion striking the coated detector surface. In a tandem mass spectrometer, the ion collection system is also capable of calculating the relative abundance of the resulting ion fragments at each particular mass. Mass spectrometers are connected to computer-based software platforms that record these mass spectra in a digital format. Subsequent data analysis allows identification of the corresponding molecular species based on their m/z ratios and relative abundance, by comparing them with a database of values for known molecules.

Mass spectrometers in use today are made up of any combination of the above-outlined ionization methods, mass analyzers, and ion detectors, and they all record the output as a set of sequential histograms, representing hits of ionized molecules on the ion detector, known as a mass spectrum (Box 11.2).

Box 11.2 The Mass Spectrum (Figure 11.5)

The mass spectrum is represented as a two-dimensional bar graph of signal intensity (on the Y -axis) versus m/z ratio (on the X -axis) containing many signal intensity peaks corresponding to the m/z ratios and intensities of the detected ions. Here m represents the mass of the ion and z represents the charge carried by the ion. The number of electrons removed is the charge number (for positive ions); +1, +2, +3 represents ions with one, two, and three charges, respectively. For ions with a charge of 1, the m/z ratio simply represents the mass of the ion. The position of a peak, or defined signal, as they are usually called, corresponds to the various m/z ratios of the ions produced from the peptides and serves as an information-rich molecular fingerprint of the peptides and proteins present in a biological specimen.

Tandem Mass Spectrometry for Peptide Identification

In combination with liquid chromatography, tandem MS (LC-MS/MS) involves multiple sequential rounds of ion selection and fragmentation in a mass analyzer (Box 11.1). Fragmentation of ions through various ion activation methods provides critical information on the molecular structure of a molecule under study (e.g. peptide sequence) and is an essential part of tandem MS. These ion activation methods, which typically are applied between different stages of mass analysis and can be used individually or in combination, result in rich fragment patterns that can provide precise information about the composition of the molecule. The speed and specificity of MS^2 data generation dictate the efficiency of LC-MS/MS for analyzing complex biological samples (e.g. the depth of sequencing of polypeptide mixtures).

Each recorded MS^2 spectrum is the result of the (often unique) fragmentation pattern produced by a particular peptide upon cleavage of its (often distinct) backbone amide and/or side chain bonds. As always in MS, peptide fragments can be detected by the ion detector only if they carry a charge. If the charge is retained at the N-terminal end of the fragment, the ion is classified as either an “a,” “b,” or “c” ion, depending on where the cleavage occurs. If the charge is retained at the C-terminal of the fragment, it is classified as an “x,” “y,” or “z” ion (Figure 11.6), with a subscript indicating the position or number of residues in the fragment. The specificity and low chemical noise of MS^2 allow for high peptide detection selectivity and sensitivity, permitting qualitative and quantitative analysis of complex

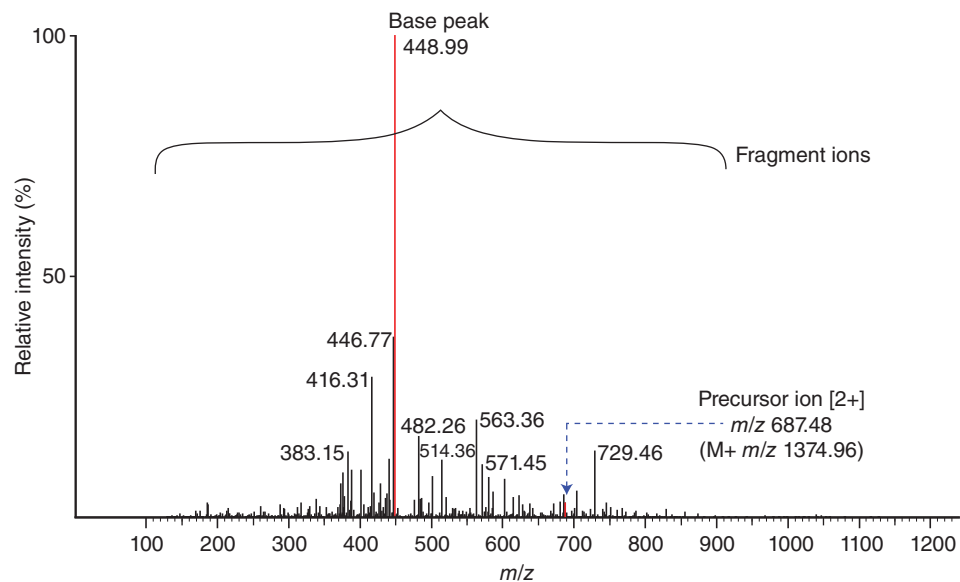


Figure 11.5 Fragmentation tandem mass spectrometry (MS/MS, or MS²) spectrum. A mass spectrum is a simple two-dimensional plot of experimentally determined ion mass-to-charge ratios versus intensity, in this case representing the distribution or pattern of product ions produced by peptide fragmentation. In the example shown, the highlighted base (most intense) peak at 448.99 *m/z* corresponds to the most abundant ion (usually set to 100% relative abundance), while other peaks represent fragment ions of a specific mass. The vertical axis shows the relative abundance or intensity, where the value displayed represents the number of ions recorded by the ion detection system (i.e. the more abundant the ion, the higher the peak). M+ is the parent molecular ion (i.e. an unfragmented peptide ion minus one electron).

protein mixtures. When combined with high-performance liquid chromatography (HPLC) or ultra-high-pressure liquid chromatography (UHPLC) peptide separations, modern MS² workflows can “shotgun” sequence thousands of distinct polypeptides in a single experiment.

The consecutive stages of mass analysis in MS² can be performed in two ways: *tandem-in-space* or *tandem-in-time*. Tandem-in-space refers to MS² instrumentation wherein two separate mass analyzers are coupled together sequentially such that *m/z* separation is done by ion selection in the first mass analyzer, followed by dissociation in an intermediate region (such as a collision chamber or ion trap); this is then followed by transmission to the second analyzer for product ion mass analysis. The second approach, tandem-in-time, uses a single mass analyzer that performs all the steps of ion selection, activation, and product ion analysis in the same device but sequentially in time. Examples of tandem-in-space instruments are combinations of quadrupole and TOF mass analyzers, while an ion trap mass analyzer can perform tandem-in-time analyses. In principle, both types of instrumentation can be expanded to allow for multiple stage MS to provide more detailed structural information, generally referred to as MS^{*n*}, where *n* denotes the number of stages of fragment analysis.

Sample Preparation

The complexity, diversity, and high dynamic range of protein concentrations in a cell, tissue, or biofluid (such as blood plasma) make the comprehensive identification and quantification of proteins challenging, especially low-abundance and membrane-associated components. To achieve better ionization efficiency and identification rates, polypeptides are typically digested into smaller peptides by enzymatic digestion with a sequence-specific enzyme like trypsin. Trypsin has exceptional cleavage specificity and cleaves the proteins by hydrolysis of the peptide bond on the carboxyl terminal side of the lysine (K) or arginine (R) residues, except when followed by proline (P); this produces peptides that are typically 6–20 or more amino

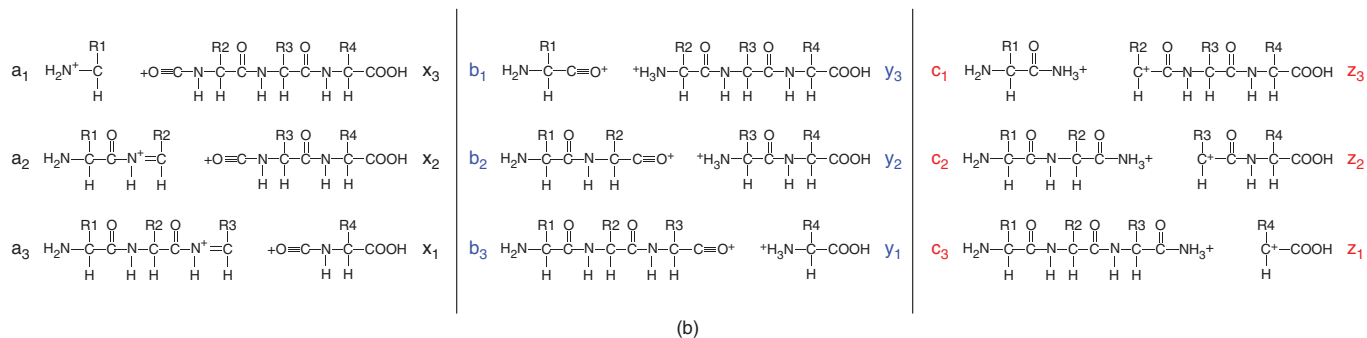
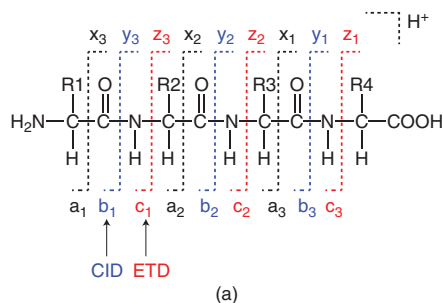


Figure 11.6 Polypeptide backbone cleavage produces different product ion species. (a) Schematic showing typical sites of polypeptide backbone fragmentation annotated with the standard Roepstorff–Fohlmann–Biemann nomenclature (Roepstorff and Fohlman 1984). Peptide fragmentation is the result of bond activation and breakage – for example, due to collision with an inert gas (CID), resulting in *b*- and *y*-ions, or upon electron transfer (ETD), resulting in *c*- and *z*-ions. Ions are labeled from the amino terminus as a_1 , b_1 , and c_1 , where the subscript represents the number of amino acid side chains contained by the ions. (b) When the charge is retained on the amino-terminal fragment, *a*, *b*, or *c* fragment ions are produced, while *x*, *y*, and *z* fragment ions are produced when the charge is retained on the carboxy-terminal fragment.

acids in length that are well suited to detection and sequencing by LC-MS/MS. Proteolytic cleavage markedly enhances detection sensitivity, improving proteome coverage. Because of its high proteolytic activity and stability under a wide variety of conditions, trypsin has become the protease of choice in MS-based proteomics, but other enzymes can provide complementary sequence coverage.

Before complex peptide mixtures can be analyzed by MS, they typically need to be processed and simplified by biochemical separation, such as by reverse-phase LC or affinity capture to enrich the peptides of interest. PTMs are of special interest, as they represent an important and common regulatory mechanism by which protein activity or associations are modulated after synthesis, either by enzyme-mediated addition of one or more covalent functional chemical groups (e.g. phosphorylation) or by proteolytic cleavage inside a living cell (Box 11.3). MS-based detection of PTMs can be highly informative biologically, as they influence virtually all aspects of normal cell biology and homeostasis, ranging from protein function through to physical binding events (e.g. protein–protein interactions). However, since PTMs are often transient and sub-stoichiometric (meaning that they do not occur on all molecules of a given protein at any one time), they can be difficult to detect. To enhance detectability, modification-specific biochemical enrichment techniques such as affinity capture have been developed to isolate PTM-modified proteins before or after digestion to aid in the detection and characterization of modified peptides. For example, selective affinity capture and analysis of serine and threonine phosphorylation is commonly achieved through the use of immobilized metal ion affinity chromatography (IMAC), such as titanium dioxide (TiO₂) beads. The different chromatographic separation methods, used alone or in combination, are aimed at producing a simplified mixture of molecules (peptides) that can be injected and ionized with higher efficiency into the mass spectrometer. In addition to facilitating detection of PTMs, sample simplification via pre-fractionation and targeted enrichment during sample preparation also serves an important role in achieving high protein sequence coverage and overall identification rates from increasingly diverse biological samples.

Box 11.3 Post-Translational Modification (Figure 11.7)

Protein post-translational modifications (PTMs) are a primary mechanism by which cells respond to environmental stimuli. They play a key role in controlling cellular processes such as signal transduction pathways regulating cell differentiation, degradation, gene expression, and protein–protein interactions. PTMs, such as phosphorylation, glycosylation, ubiquitination, nitrosylation, methylation, acetylation, sumoylation, and proteolytic processing, routinely influence virtually all aspects of normal cell biology and homeostasis. As PTMs are often dynamic, sub-stoichiometric (incomplete), and transient (reversible), they contribute to an exponential increase in the functional and structural diversity of the proteome. Identifying and understanding their role is critical in the study of cell biology, disease pathogenesis, and developing new treatments.

Phosphorylation is the most frequently occurring and studied PTM and there are more than 58 000 experimentally determined modification sites with experimental evidence, making “phosphoproteomics” an important sub-branch of functional proteomics. Phosphorylation, a key reversible modification, occurs by the addition of a phosphate group at the serine, threonine, or tyrosine residues (also histidine in prokaryotes) of the protein and plays a critical role in maintaining the integrity of the myriad cellular processes and signaling pathways in the cell. For example, plasma membrane receptor-associated protein kinases are enzymes that catalyze the phosphorylation of tyrosine residues of critical intracellular signaling proteins that play an important role in the signal transduction process. Genomic aberrations that disrupt the role of tyrosine kinases can lead to cell transformation and cancer, as seen with the tyrosine kinase protein ABL. ABL mutations leading to the formation of a BCR–ABL1 fusion protein drive the pathogenesis of chronic myelogenous leukemia (CML), a curable cancer of the bone marrow that can be effectively targeted by inhibitory drugs.

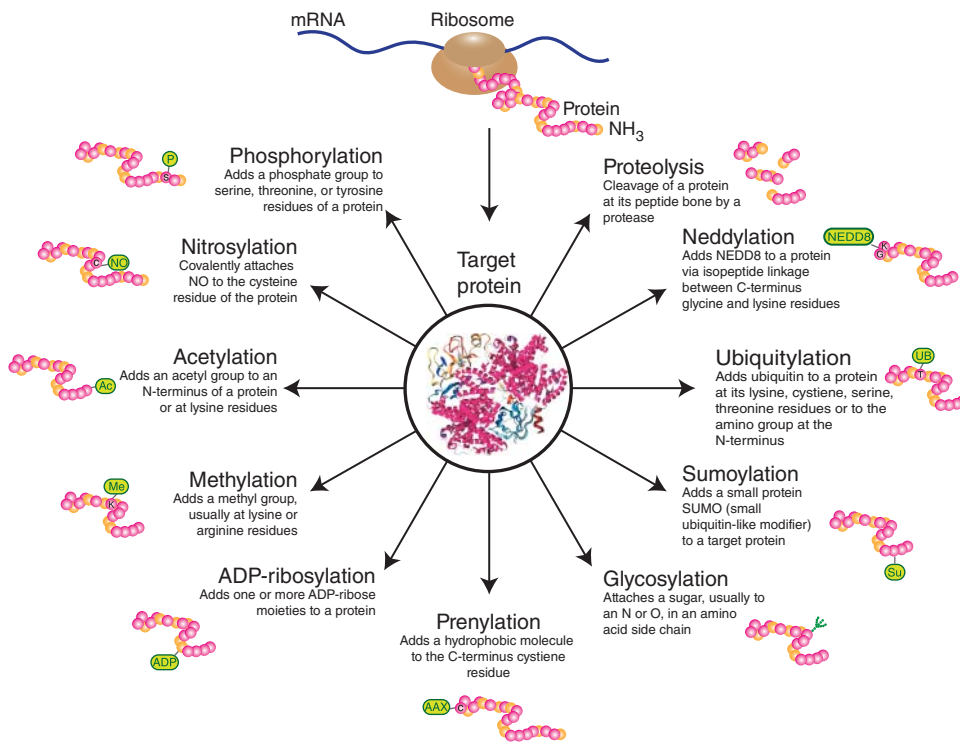


Figure 11.7 Post-translational modifications (PTMs) take place at different amino acid residues in proteins. While there are currently >50 PTMs listed in the UniProt database, the figure lists some of the better studied PTMs.

Bioinformatics Analysis for MS-based Proteomics

The large volume of data obtained from an MS experiment, comprising tens of thousands of data points in virtually every spectrum, is inherently noisy because of measurement errors, missing values, and artifacts introduced during different phases of the experiment. Before the spectra can be used for the identification of true signals such as peptide fragments, the data need to be cleaned or pre-processed through the use of multi-variate statistics; this reduces spectrum noise and complexity (dimensionality) to generate a much smaller and statistically manageable set of defined peaks prior to peptide or protein identification. Most commercial MS instruments include software that performs data pre-processing based on various pre-specified parameters and algorithms to facilitate a variety of signal manipulations that include baseline correction, smoothing, normalization, and peak-picking to produce more easily interpreted MS spectra (Figure 11.8a). Data smoothing applies signal processing techniques like Savitzky-Golay filtering, mean or median filtering, or Gaussian filtering to remove low signal fluctuations present in the spectra due to instrument-derived noise. Baseline correction involving methods like Top Hat filter, Loess derivative filters, or linear splines (Bauer et al. 2011), enabling the removal of estimated chemical noise that can arise due to trace contaminants present throughout the instrumentation workflow. While smoothing and baseline correction is applied to each spectrum individually, normalization corrects for systematic instrument variability by converting all spectra to the same intensity range to make the spectra recorded within an experiment more comparable. A final crucial step is the definition of peaks, or *peak picking*, which involves determining the precise mass, apex, and intensity for each peak using one of several established methods based on signal-to-noise ratio (SNR), centroid, Gaussian fit, or center-of-width at half-maximum height metrics. The resultant peak list is subsequently used in further downstream statistical analysis and biological interpretation.

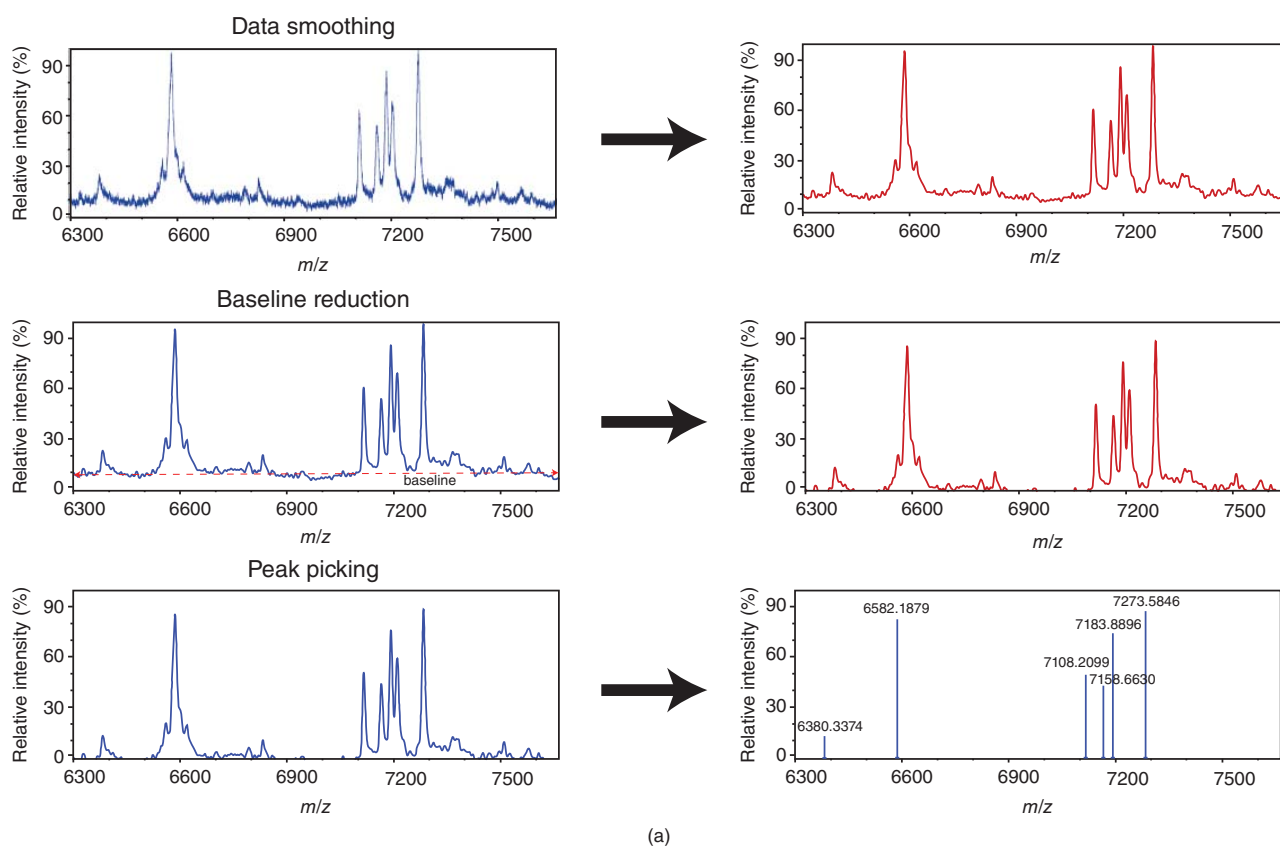
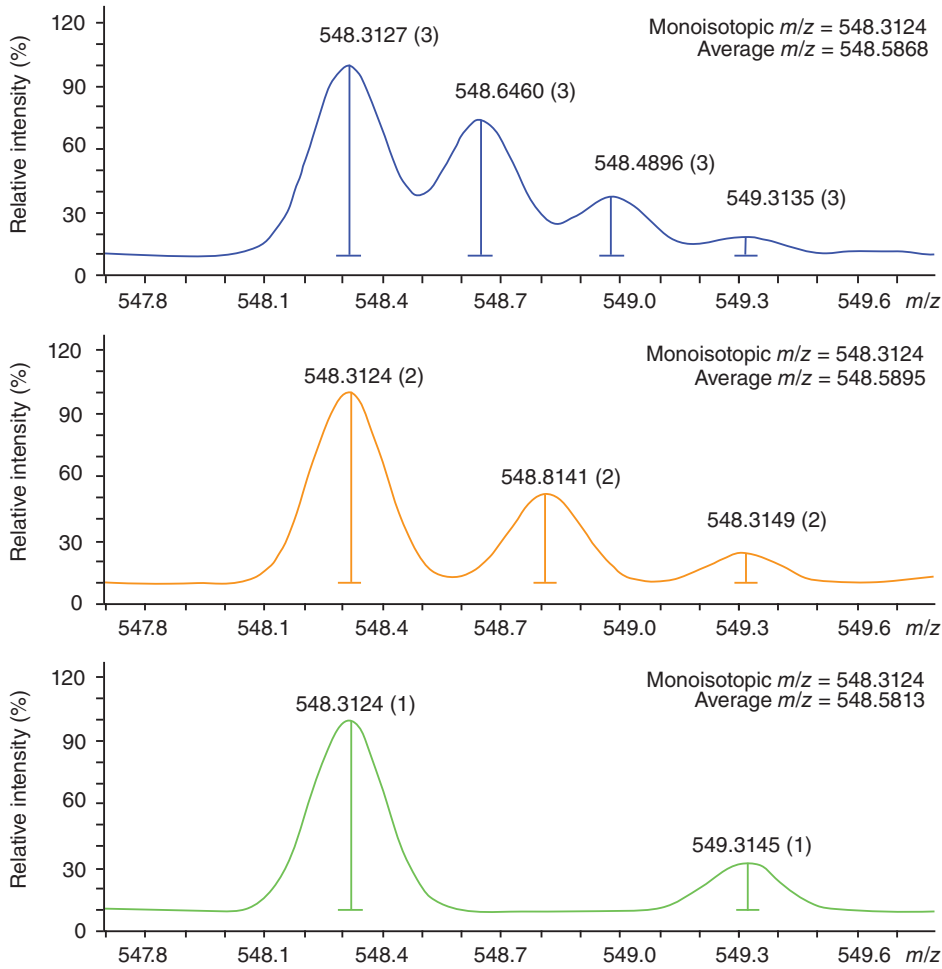
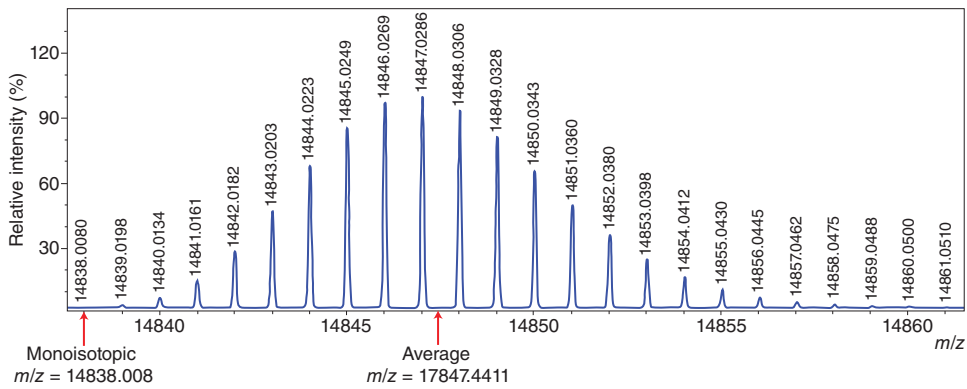


Figure 11.8 Data pre-processing workflow of a mass spectrum. Different steps in the pre-processing workflow of a mass spectrum. (a) Pre-processing steps include data smoothing, baseline correction, and peak picking. Numerous algorithms are available for each step in the transformation of a mass spectrum into a peak list suitable for further statistical analysis in the process of peptide and protein identification. (b) Illustration of a possible mass spectrum of a peptide fragment with a monoisotopic mass of 584.3124 at different (+1, +2, +3) charge states. (c) The isotopic envelope of *Gallus gallus* protein lysozyme illustrating its monoisotopic and average mass.



(b)



(c)

Figure 11.8 (Continued)

Several advanced processing techniques have been developed to define clusters of related peptide peaks arising from the presence of multiple charge states, the natural prevalence of stable isotopes, and mass shift arising from PTMs. Since mass spectrometers measure m/z ratios rather than mass per se, ions with identical mass but multiple charge states (e.g. +1, +2, +3, stemming from the presence of one, two, or three proton ions) are detected with different m/z ratios. For example, a peptide ion bearing a +2 charge (doubly protonated) versus a single (+1) charge will differ in the detected m/z ratio by roughly half, while the third ion with a +3 charge (triply protonated) will present with only one-third the m/z value, and so on (Figure 11.8b). Whereas MALDI ionization usually produces ions with a low (+1) charge state, the ESI process frequently produces precursor ions with multiple charge states. For the purpose of accurate detection, it is desirable to transform each m/z spectrum to a mass representation independent of charge state where all the multiply charged forms of a given peptide that are detected by MS are recalculated into the corresponding singly charged form and grouped together to account for total intensity and peak width. This process of reducing multiply charged states to a single mass measurement is called *charge state reduction* or *deconvolution*. These processes involve the use of software tools that exploit the high resolution of modern mass spectrometers that are capable of resolving the distinct stable isotopic peaks for individual peptides. For example, a peptide with a single ^{13}C (heavy isotope) element will be measured as 1 Da heavier than its corresponding ^{12}C (bulk natural carbon) counterpart. As most biological molecules natively display this kind of isotopic variation (here, as $\sim 1\%$ of all carbon is ^{13}C), multiple isotopic peaks are typically observed for each molecule, producing an ion envelope that exhibits a characteristic mass shift in m/z ratios (Figure 11.8c).

Mass spectrometer systems produce data in two forms, as follows.

- *Average mass*. This is simply the weighted average mass of all observed isotopic forms of the molecule and usually reported by low-resolution instruments that are unable to resolve isotopes.
- *Monoisotopic mass*. This is calculated from high-resolution spectra as the sum of the exact masses of the most abundant isotope of each element through the process of “de-isotoping” to remove unwanted isotopes from the final peak list.

Monoisotopic mass is considered more accurate, given that average mass cannot be determined as precisely because of variations in natural isotopic abundances. The detection of isotope peaks and the monoisotopic mass also aid the process of charge deconvolution. For example, a molecule with a +2 charge will have stable isotope peaks that are separated on the spectrum by ~ 0.5 (1/2) Da apart, +3 ~ 0.33 Da apart, and so forth.

Mass shifts can also result from the presence of chemical adducts (like sodium) that associate with a peptide in vitro, chemical modifications such as biological PTMs, or experimentally induced alterations in vitro. For example, during sample preparation, it is common for methionine residues to become oxidized, adding 16 Da in mass for each oxygen atom added. Therefore, adduct and PTM detection relies on defining the alteration to the mass of a tryptic peptide and the product fragments resulting from the modification of a specific amino acid residue side chain. For accurate identification of the modification site, it is necessary to detect a characteristic mass shift in both the precursor peptide ion and the subset of N- and C-terminal fragment ions carrying the modified residue. With good MS^2 data quality from high-resolution instruments, it is possible to reliably identify and localize one or more putative modified residues of individual peptides.

Proteomics Strategies

The two major proteomics strategies employed in the analysis of proteins are the “bottom-up” and “top-down” approaches. The analysis of peptides obtained from proteolytic digestion of proteins is generally referred to as bottom-up or shotgun proteomics and has formed the

basis for the majority of proteomics research undertaken to date. As opposed to the bottom-up approach, top-down proteomics (TDP) is a concept that applies MS to explore the “proteoforms” of intact proteins. Bottom-up strategies may involve either a targeted or a global approach. In targeted proteomics, only a small, pre-selected group of proteins is analyzed exclusively by MS whereas, in global proteomics, one attempts to analyze all the proteins present in a given sample with minimal bias.

Most standard bottom-up MS-based proteomics studies have three distinct stages (Figure 11.9).

- 1) Extraction and purification of proteins from one or more biological sources using various biochemical methods, followed by proteolytic digestion of the isolated proteins into peptides and further liquid chromatographic fractionation of the resulting mixture.
- 2) Qualitative and/or quantitative mass spectrometric analysis of the resulting peptides.
- 3) Computational analysis of the recorded spectral datasets based on sequence database searches to determine peptide amino acid sequences, with the goal of both identifying and quantifying proteins followed by statistical analysis to ensure confident assignments.

Proteomics studies can differ in their scientific objectives and can be either qualitative or quantitative. Qualitative studies focus on the systematic identification of proteins in a sample and the characterization of their PTMs, whereas quantitative proteomics aims to measure absolute or relative protein levels such as differences in protein abundance between samples (e.g. case versus control; Box 11.4). Quantitative proteomics is a powerful strategy for use in both shotgun and targeted analyses to understand global protein expression dynamics and changing PTM patterns in a cell, tissue, or organism under different conditions (such as pathophysiological contexts) through the quantification of cognate molecular ions. This approach has found a productive niche in the contexts of systems biology, biomarker discovery, and biomedical research.

Box 11.4 Quantitative Proteomics (Figure 11.10)

- *Label-free quantification.* This is a relative quantification technique used to compare protein or peptide levels in between two or more liquid chromatography tandem mass spectrometry (LC-MS/MS) runs. Here, the assumption is that, under ideal conditions, identical peptides measured across different experimental conditions can be compared directly using the recorded MS¹ intensities or spectral counts. The advantages of the label-free technique are that it does not require the extra experimental steps needed for labeling and any number of experiments can be readily compared. A disadvantage stems from the under-sampling problem inherent to MS/MS, in which not all peptides present in a complex mixture are consistently detected between samples, even replicate runs, leading to variance in abundance estimates that dampens statistical measures of differential levels.
- *Labeling strategies.* Proteomics samples can be isotopically labeled through metabolic labeling in vivo or by in vitro chemical tagging of the extracted proteins or peptides. Since the incorporated light and heavy isotopic forms of a (poly)peptide are chemically identical, they typically co-elute together during LC fractionation and so can be detected concurrently, yet distinguished according to their different masses evident during MS analysis. Subsequently, the ratio of peak intensities recorded for heavy and light labeled peptides measured across two or more experimental groups can then be compared to determine changes in abundance in one sample relative to that of the other(s). If the measurements are generated in a precise manner, statistically significant changes can be reliably deduced. Various isotope labels or tags can be introduced at either the

(Continued)

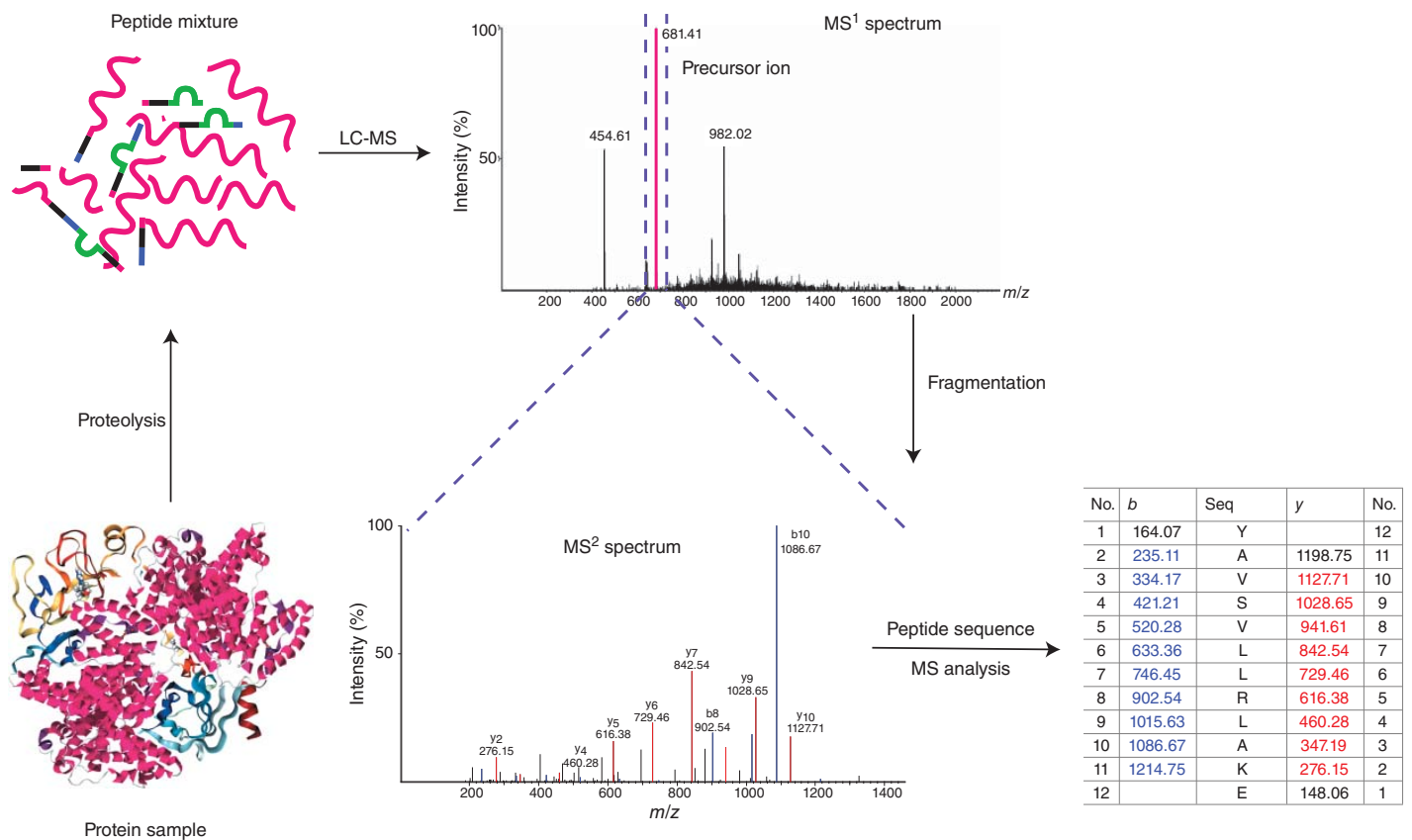


Figure 11.9 Shotgun proteomics workflow. Schematic showing different steps involved in a typical “bottom-up” liquid chromatography tandem mass spectrometry (LC-MS/MS) pipeline. A protein sample is enzymatically digested, usually using trypsin, into peptides, which are then subject to chromatographic separation to simplify the sample before injection (electrospray) into a tandem mass spectrometer for fragmentation. After ionization, gas-phase precursor ions are scanned in the first round to give MS¹ spectra. These parent (precursor) ions are either individually (data-dependent acquisition) or concomitantly (data-independent acquisition) fragmented to give MS² spectra. The acquired MS² spectra are then analyzed via various search algorithms for peptide identification (database or library searching) and protein inference (integrative scoring).

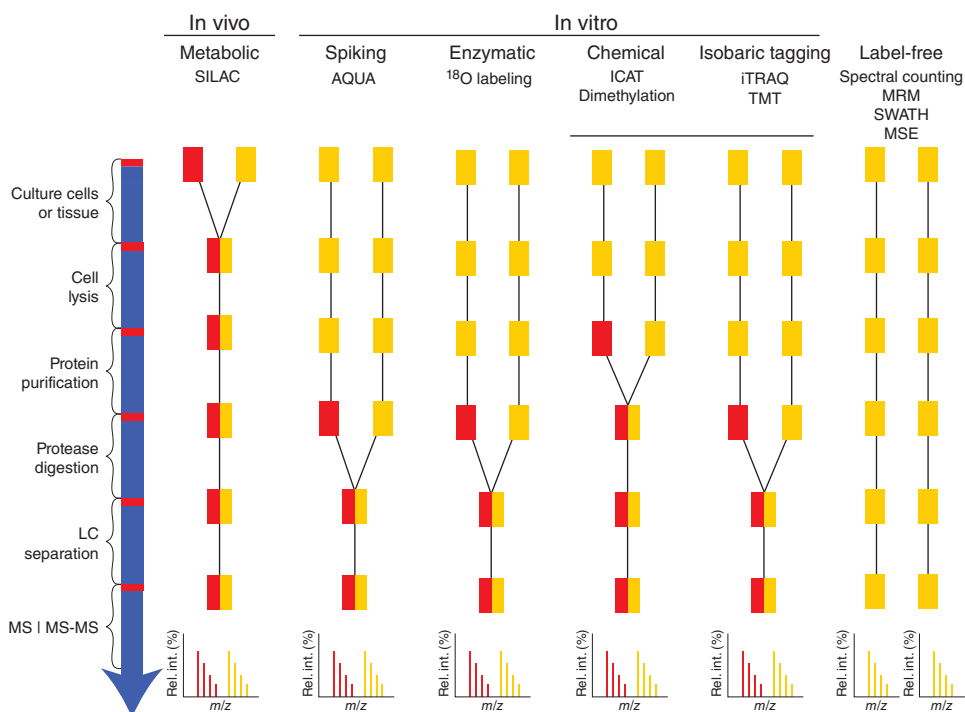


Figure 11.10 A schematic diagram comparing the label-free approach with the different labeling strategies. The isotopic label represented by the red rectangle is introduced into the sample during various different stages in the quantitative proteomics workflow, pooled and subjected to mass spectrometer analysis.

protein or peptide level during sample preparation using in vivo and in vitro methods. Different labeling strategies are discussed below.

- *Metabolic labeling.* Stable isotope labeling in vivo is done by growing the cells or organisms under study in the presence of defined amino acids or nutrients consisting of one or more heavy isotopes. One of the more popular metabolic labeling techniques is stable isotopic labeling by amino acids in a cell culture (SILAC; Ong et al. 2002), where heavy isotopes present in the culture medium are introduced into mammalian cells during growth, causing a predictable shift in the mass of digested peptides during MS analysis proportional to the incorporation efficiency of the label. Peptides in the differentially labeled samples analyzed by MS are typically detected as paired peaks, where the mass difference observed reflects both the number and nature of the labeled amino acids used, allowing rapid comparison of peptide and protein ratios. Heavy labeled lysine and arginine are used in SILAC experiments, for double (or triple) labeling of two (or three) samples under comparison. Other isotope labeling techniques primarily use in vitro methods and, as described below, are usually carried out via covalent modification by means of chemical or enzymatic processing of purified or proteolytic digested test and reference protein samples.
- *Chemical labeling.* Isotope-coded affinity tagging (ICAT; Gygi et al. 1999) is a pioneering chemical labeling technique in which protein samples are coupled with either an isotopically heavy or light reagent at cysteine residues. The ICAT reagent is made up of three elements: a cysteine directed reactive group for labeling the amino acid side chain, an eightfold deuterated (d8; which adds 8 Da to the molecular mass of the peptide) or light (d0) linker region, and a biotin tag for affinity isolation of labeled

(Continued)

Box 11.4 (Continued)

polypeptides. The labeled samples are then pooled, digested with an appropriate protease such as trypsin, subjected to tag capture on streptavidin affinity columns, and then eluted before undergoing MS analysis.

An alternative and less costly chemical labeling technique is dimethyl labeling, which is applied after the proteolytic digestion of proteins and is based on the reaction of peptide primary amines (peptide N-termini and the epsilon amino group of lysine residues; Hsu et al. 2003). The dimethyl labeling reagent is composed of formaldehyde and cyanoborohydride, with labeled forms containing a combination of deuterated hydrogen and ^{13}C atoms and can be used in triplex. This makes it possible to quantitatively analyze three samples in a single MS run by comparing the mass difference of the dimethyl labels to determine protein abundance in different samples.

- *Enzymatic labeling.* Enzymatic labeling techniques, such as proteolytic labeling using a protease like trypsin, can introduce ^{18}O (or regular ^{16}O) labeled water during the cleavage (hydrolysis) reaction, generating isotopically labeled peptides. For example, in a two-step reaction, an ^{18}O or ^{16}O atom is incorporated into the carboxyl terminus of the resulting peptide upon hydrolytic cleavage of a polypeptide, followed by a carboxyl oxygen exchange reaction which incorporates a second ^{18}O (or ^{16}O) atom into the carboxyl terminus of each peptide (Miyagi and Rao 2007).

All isotope labeling techniques allow for relative quantification based on measuring the mass difference between differentially labeled peptides but are limited by the number of samples that can be studied (multiplexed) together in one experimental group. The mass difference concept is generally restricted to a binary (2-plex) or ternary (3-plex) set of reagents, as higher order multiplexing leads to increasing complexity and diminished discrimination in the MS^1 spectra. This limitation can be overcome to a certain extent by the use of isobaric tags (see Isobaric Tagging) that are designed for higher multiplexing.

- *Isobaric tagging.* The isobaric tag for relative and absolute quantification (iTRAQ; Ross et al. 2004) and tandem mass tag (TMT) reagents (Thompson et al. 2003) represent two isobaric labeling techniques available for quantitative MS. Reagents for iTRAQ are available in 4-plex and 8-plex forms, while TMT reagents are available in 2, 4, 6, 8, 10, and, most recently, 11-plex forms. These isobaric stable isotope tags are composed of a mass reporter that has a unique number of ^{13}C and/or ^{15}N heavy isotope substitutions, followed by a mass normalizer that balances the mass of the tag to make all tags equal in mass so they form a common m/z peak during MS^1 precursor ion scans, and lastly a reactive region that cross-links the label to either amine or cysteine residues on the target polypeptides. Samples are labeled with individual mass tags and combined for LC-MS/MS analysis. Since all tags have the same mass, identical peptides present in all samples display the same mass shift and elute together as a single precursor ion peak by MS^1 . After MS^2 fragmentation (e.g. through higher energy collisional activation dissociation-based collisional activation), the reporter tags are cleaved off simultaneously at a specific linker region to form molecular fragment reporter ions with close but distinct masses, allowing for parallel quantification of relative peptide intensities, together with the corresponding peptide fragment ion series suitable for protein sequence identification.

There are two major approaches for comparative quantitative proteomics analyses: isotope labeling versus label-free techniques. The MS^2 methods used in quantitative proteomics are largely the same as those used for protein identification but include an additional dimension for making abundance measurements. In the label-free approach, each sample is analyzed

separately by LC-MS/MS, and the measured ion levels are then compared based on the MS² spectral counts obtained per protein or PTM site in each specimen or sample set. In a label-based approach, samples in an experimental group are isotopically labeled (e.g. via chemical tagging in vitro or by in vivo metabolic labeling), then combined and analyzed together in the same MS run, with the distinct masses of the isotopic labels distinguishing both the source of the multiplexed samples and the corresponding relative levels of proteins present in each specimen. Isotopic labeling strategies are generally considered more accurate, since samples are compared directly, and can produce more reproducible results, as variation due to sample processing and MS under-sampling is minimized; however, these strategies are more costly and time-consuming to implement, more limited in terms of the number of samples that can be pooled, and require specialized software tools for data analysis (see PSM Software). On the other hand, while label-free approaches scale well with respect to the total number of samples being analyzed and are easier to implement, they can be less effective at detecting small differences in protein abundance and can suffer from lower reproducibility.

Relative quantification compares protein or peptide levels between samples in two experimental groups – such as measuring differences in the molecular profiles in healthy versus diseased states, mutant versus wild-type cells, or progenitor versus differentiated cells (Filiou Michaela et al. 2012) – while absolute quantification uses reference standards to determine the precise quantities of one or more target protein or peptide in one or more samples, making it useful for determining protein concentrations, protein complex subunit stoichiometry, and the extent of PTMs (Gerber et al. 2003).

In addition, there are two different strategies commonly used in proteomics that encompass targeted MS: *discovery-based global profiling approaches* and *hypothesis-driven directed approaches* (Schubert et al. 2017). Discovery-based proteomics studies are open-ended and can be performed when using bottom-up shotgun sequencing via data-dependent acquisition (DDA) procedures in which all peptide ions above a pre-determined intensity are selected for MS² fragmentation and subsequently identified from the resulting fragmentation spectra in an iterative (serial) manner. It is also possible to conduct data-independent acquisition (DIA), where the co-fragmentation and identification of peptides in a sample is done in a more systematic, multiplexed manner, most notably by analyzing all peptides simultaneously across a certain mass range. In comparison, hypothesis-driven proteomics uses prior information to pre-select only one or a few proteins and peptides of particular interest for MS² analysis. This includes targeted MS detection such as selected reaction monitoring (SRM), where the signal intensity patterns of a few pre-defined fragment reporter ions specific to a target protein or peptide of interest are selectively screened by MS to confirm molecular identity. Selective detection of these patterns in spectra identifies the corresponding molecule. The advantage of targeted proteomics is that the selective screening allows for more sensitive and specific protein detection. The detection of peptides bearing specific PTMs, representing a well-defined molecular response, or candidate circulating biomarkers present in trace levels in the blood are examples of where targeted proteomics is most commonly used.

The shotgun approach, though popular and relatively simpler to implement, has to contend both with ambiguity in terms of protein inference (peptide-to-protein assignments) and with incomplete and inconsistent sequence/modification coverage. The connectivity between a parental intact protein and the corresponding digested peptides is lost during the bottom-up workflow, leading to complications during integrative analysis (i.e. the assignment problem). As the TDP approach measures both the intact protein and fragment ion masses produced by MS², higher sequence coverage with fewer gaps can be achieved, facilitating characterization of protein variants such as proteoforms. Yet, while the TDP technique minimizes inference problems, it is highly dependent on feature discrimination and deconvolution (e.g. resolving highly complex multiply charged intact protein ion envelopes produced by ESI; Kelleher et al. 1999).

Now that readers have been introduced to the general concepts behind protein MS, sample preparation, and basic data analysis by biological MS, we can examine the more popular MS-based proteomics techniques in use today.

Peptide Mass Fingerprinting

Peptide mass fingerprinting (PMF) is a conceptually simple protein identification technique in which a single polypeptide (e.g. a gel band) is cut into smaller peptides with a sequence-specific protease (typically trypsin), followed by accurate MS determination of the resulting peptide masses. MALDI or ESI analysis provides a fast, accurate, and efficient way to identify proteins in gel bands or spots. The premise of PMF is that any unique protein can be readily described as a set of unique peptide masses that correspond to the amino acids constituting specific sub-sequences generated by enzymatic cleavage. While certain proteins may be highly similar (encoded by gene duplicates/paralogs), some portion of a protein's sequence is typically unique and therefore should produce specific, identifiable combinations of peptide masses. Hence, if a particular polypeptide is cleaved in a specific manner, then the resulting peptide masses obtained by MS form a unique "fingerprint" that maps back specifically to the corresponding protein sequence, known in advance (i.e. obtained from a reference sequence database).

Key to the PMF protein identification process is the comparison and matching of experimentally determined peptide masses with theoretically predicted masses. Peptide masses can be inferred *in silico* by taking annotated protein sequences (from a given organism) and computationally cleaving them using the same enzyme (e.g. trypsin) rules used to process the real samples. The mass of each peptide is calculated for each protein in the database, and the pattern compared with the observed masses from the PMF analysis (Figure 11.11). Statistical methods are used to determine which combination of theoretical peptides for a given protein best match the observed peptides; this usually includes performing a significance assessment to calculate a probability that the match occurs by chance (i.e. a false positive), with the best correspondence within a pre-defined mass error range (mass tolerance) deemed the most likely candidate. Obviously, PMF is more error prone with protein mixtures and cannot be used if the organism under study has not been sequenced. Also, care must be taken in sample handling to avoid the presence of irrelevant peptides originating from contaminants such as from hair and skin or the autolysis of trypsin, as these can lead to spurious results. Protein digestion is a stochastic process, and a protease may not fully cleave a polypeptide at every occurrence of the cleavage site, resulting in missed cleavages. Incomplete proteolytic digestion can result in long(er) peptides that are harder to detect or fragment. A theoretically digested sequence database with all possible partially cleaved peptides will also result in an exponential increase in complexity. Spurious results can also result if not accounting for the presence of unknown PTMs or chemical modifications during sample preparation (e.g. oxidation) that can increase or decrease molecular weight. The extent of modifications can either be incomplete (variable modifications) or ubiquitous on all occurrences of a particular amino acid (fixed modification); for example, carbamidomethylation of cysteine, a reaction commonly used during sample preparation in order to prevent cysteine cross-bridges after sample digestion, which increases the molecular weight of cysteine. As only peptide masses (and not exact sequences) are matched, the presence of PTMs can lead to ambiguous results. The theoretical number of peptide masses in the database grows exponentially with each variable modification, resulting in reduced match specificity and a large increase in search time. Hence, to reduce computational complexity, the number of variable modifications allowed needs to be restricted while, at the same time, the reference database needs to account for all possible combinations of missed cleavages and variable modifications.

The PMF database search concept was first implemented by Henzel and colleagues, who developed the Fragfit computational algorithm in 1993 (Henzel et al. 1993). This program was

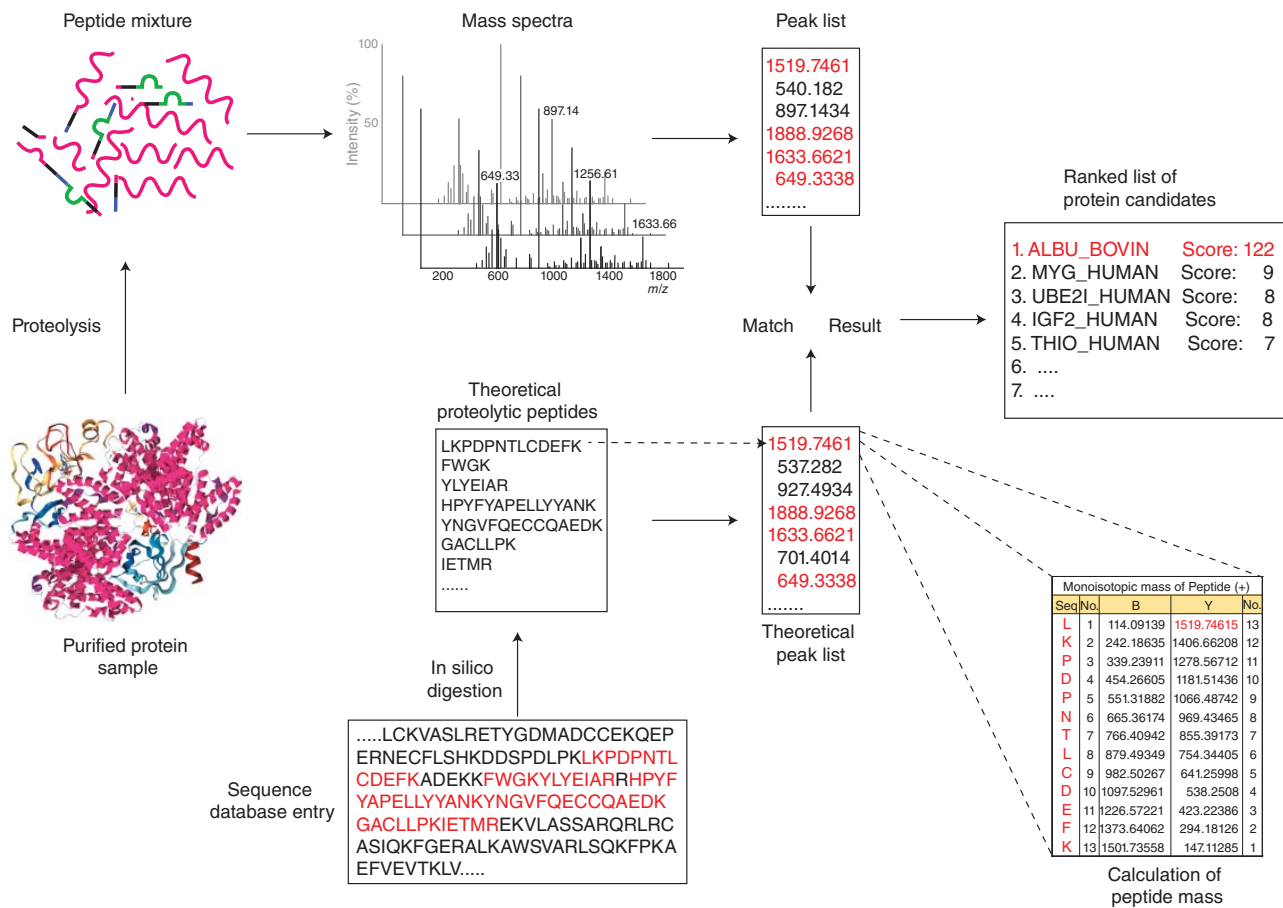


Figure 11.11 Peptide mass fingerprinting (PMF) workflow. Schematic showing different steps in a PMF pipeline. A purified protein sample is enzymatically digested and extracted peptide masses are measured by mass spectrometry (MS), typically by matrix-assisted laser desorption ionization (MALDI) time of flight (TOF) MS because of its speed and simplicity. The observed peptide mass patterns are then compared with theoretically calculated masses derived by in silico application of enzyme cleavage rules to protein sequences in a reference database and analyzed using a search algorithm to obtain a ranked list of protein candidates.

used to accurately identify proteins isolated from an *Escherichia coli* cell lysate using only three peptide masses per protein while searching a database consisting of 91 000 candidate protein sequences, showcasing the use of computational software in conjunction with MS for protein identification. While the central advantage of the PMF method is that only the masses of (unique) peptides must be (accurately) measured, PMF algorithms are confounded by mixtures containing more than one protein. Ideally, if the sequence of the peptides could be determined, rather than just their masses, then the problem with protein mixtures and confidence in protein identifications could be improved. As documented below, peptide sequencing is actually possible by tandem mass spectrometric techniques.

PMF on the Web

Mascot

Mascot, a widely used commercial MS search engine, was one of the first search engines to incorporate a probability-based scoring approach for peptide and protein identifications. It was derived from the MOWSE probability algorithm, which was capable of only conducting PMF searches and required pre-indexed enzyme-specific databases prior to computing peptide mass values. This made it difficult to search for potential PTMs as a new database was needed for each combination of modifications. Mascot was developed to overcome these limitations by computing mass values directly from sequence databases “on the fly,” thus removing the need for database indexing and adding additional support for PTMs and more flexible search strategies (Perkins et al. 1999).

Data are submitted online in the form of peptide masses or peak lists. Other search-specific parameters (see Table 11.2) include sample (species) taxonomy, protein reference database, potential modifications, proteolytic enzyme, the number of missed cleavages allowed, MS scan mode used, and whether to use monoisotopic or average mass values to calculate the peptide mass; these, together with the protein mass window and match error tolerance, are provided as input. At the end of a search, a detailed summary report is generated encapsulating the putative peptide and protein identifications (Figure 11.12).

Mascot’s basic approach in peptide identification is to calculate the probability of observing a match between the observed experimental data and a candidate from the reference database by chance alone, with a peptide showing the lowest probability deemed the best match and is reported as $-10 \cdot \log_{10}(P)$, where P is the actual probability. Mascot also estimates significance by calculating the false discovery rate (FDR; see Box 5.4) using a target–decoy approach in which the search is repeated with the same parameters against a database where the sequences are reversed or randomized. Since no true matches are expected from the “decoy” database, the number of matches here can be used as a good estimate of the number of false positives in the results. As the aim of any spectral matching probability-based scoring algorithm is to assign a level of confidence to a peptide-spectrum match to weed out false positives, this approach represented a huge advantage over other MS search tools of its time.

Proteomics and Tandem MS

While PMF is a simple and fast method of protein identification, it suffers from some important drawbacks. The presence of multiple different proteins, unaccounted splice variants, and PTMs causing unexpected mass shifts hinders its effectiveness. Only proteins with peptides within the recorded mass range corresponding to defined sequences in a database can be identified. The introduction of tandem MS (MS/MS or MS²) helped to overcome many of these limitations. Additional information gained from the secondary peptide fragmentation and the better search algorithms has made the analysis of complex protein mixtures possible.

MASCOT Peptide Mass Fingerprint

Your name **Email**

Search title

Database(s) **Enzyme**

Allow up to missed cleavages

Taxonomy

Fixed modifications

Variable modifications

Protein mass kDa **Peptide tol. ±** Da

Mass values MH⁺ M_r M-H⁻ **Monoisotopic** **Average**

Data file No file chosen

Query

Data input

Decoy **Report top** hits

Matched peptides (red):
 mTRAQ:13C(3)15N(1) (K)
 mTRAQ:13C(3)15N(1) (N-term)
 mTRAQ:13C(3)15N(1) (Y)
 mTRAQ:13C(6)15N(2) (K)
 mTRAQ:13C(6)15N(2) (N-term)
 mTRAQ:13C(6)15N(2) (Y)
 NIPCAM (C)
 Oxidation (HW)
 Phospho (ST)
 Phospho (Y)
 Propionamide (C)

(a)

Figure 11.12 Mascot peptide mass fingerprinting (PMF). PMF submission screen and search results showing the representative protein summary report indicating matched peptides (red). (a) PMF submission form where search-specific parameters such as an enzyme, number of missed cleavages, organism taxonomy modifications, and peptide masses can be selected. (b) Proteins search results page displays a ranked list of proteins each with a $-\log_{10}(P)$ protein score. The protein with the highest significance score is considered the most likely match. (c) The protein view page displays the peptides identified (red) in the matched protein sequence, percent sequence coverage, and the number of mass values searched and identified. (d) The continuation of the protein view page lists the position of identified peptide sequences along with experimental masses (entered in the submission form) and the calculated and theoretical peptide masses from the protein sequence database used in the search.

MATRIX SCIENCE Mascot Search Results

User : abc
 Email : abcd@xyz.com
 Search title : Sample1
 Database : SwissProt 2018_03 (557012 sequences; 199714119 residues)
 Taxonomy : Mammalia (mammals) (66920 sequences)
 Timestamp : 26 Apr 2018 at 21:26:59 GMT
 Top Score : 61 for **ALBU_BOVIN**, Serum albumin OS=Bos taurus OX=9913 GN=ALB PE=1 SV=4

	SwissProt	Decoy
Protein hits above identity threshold	0	0
Highest scoring protein hit	61	51

Mascot Score Histogram

Protein score is $-10 \cdot \log(P)$, where P is the probability that the observed match is a random event. Protein scores greater than 61 are significant ($p < 0.05$).

The histogram displays the number of hits for various protein scores. The x-axis represents the protein score, and the y-axis represents the number of hits. The distribution shows a peak at a score of 61, which is highlighted in red, indicating a significant match. Other scores are shown in green with diagonal hatching.

Concise Protein Summary Report

Format As: Concise Protein Summary [Help](#)

Significance threshold $p < 0.05$ Max. number of hits: AUTO

Preferred taxonomy: All entries

Re-Search All Search Unmatched

1. [ALBU_BOVIN](#) Mass: 71244 Score: 61 Expect: 0.058 Matches: 38
 Serum albumin OS=Bos taurus OX=9913 GN=ALB PE=1 SV=4

(b)

Figure 11.12 (Continued)



MASCOT Search Results

Protein View: ALBU_BOVIN

Serum albumin OS=Bos taurus OX=9913 GN=ALB PE=1 SV=4

Database: SwissProt
 Score: 61
 Expect: 0.058
 Monoisotopic mass (M_r): 71244
 Calculated pI: 5.82
 Taxonomy: [Bos taurus](#)

Sequence similarity is available as [an NCBI BLAST search of ALBU_BOVIN against nr.](#)

Search parameters

Enzyme: Trypsin: cuts C-term side of KR unless next residue is P.
 Fixed modifications: [Carbamidomethyl \(C\)](#)
 Variable modifications: [Oxidation \(M\)](#)
 Mass values searched: 81
 Mass values matched: 38

Protein sequence coverage: 48%

Matched peptides shown in **bold red**.

```

1 MKWVTFISLL LFFSSAYSRG VFRRDTHKSE IAHRFKDLGE EHFKGLVLIA
51 FSQYLQQCPF DEHVKLVNEL TEFAKTCVAD ESHAGCEKSL HTLFGDELCK
101 VASLRETYGD MADCCEKQEP ERNECFLSHK DDSPDLPKLK PDPNTLCDEF
151 KADEKKEFWGK YLYEIARRHP YFYAPELLYY ANKYNGVFQE CCQAEDKGAC
201 LLPKIETMRE KVLASSARQR LRCASIQKFG ERALKAWSVA RLSQKFPKAE
251 FVEVTKLVTD LTKVHKECCH GDLLECADDR ADLAKYICDN QDTISSKLKE
301 CCDKPLLEKS HCIAEVEKDA IPENLPPLTA DFAEDKDVCK NYQEAKDAFL
351 GSFLYEYSRR HPEYAVSVLL RLAKEYEATL EECCAADDPH ACYSTVFDKL
401 KHLVDEPQNL IKQNCQDFEK LGEYGFQNAL IVRYTRKVPQ VSTPTLVEVS
451 RSLGKVGTRC CTKPESERMP CTEDYLSLIL NRLCVLHEKT PVSEKVTKCC
501 TESLVNRRPC FSALTPDETY VPKAFDEKLF TFHADICTLP DTEKQIKKQT
551 ALVELLKHKP KATEEQKTV MENFVAFVDK CCAADDKEAC FAVEGPKLVV
601 STQTALA

```

Unformatted sequence string: [607 residues](#) (for pasting into other applications).

Sort by residue number increasing mass decreasing mass
 Show matched peptides only predicted peptides also

(c)

Figure 11.12 (Continued)

Start - End	Observed	Mr (expt)	Mr (calc)	Delta M	Peptide
66 - 75	1163.6306	1162.6233	1162.6234	-0.0000 0	K.LVNELTEFAK.T
89 - 100	1418.7381	1417.7308	1418.6864	-0.9556 0	K.SLHTLFGDELCK.V
101 - 105	545.3405	544.3332	544.3333	-0.0001 0	K.VASLR.E
131 - 138	886.4152	885.4079	885.4080	-0.0001 0	K.DDSPDLPK.L
152 - 156	590.3144	589.3071	589.3071	0.0000 1	K.ADEKK.F
152 - 160	1108.5785	1107.5712	1107.5713	-0.0001 2	K.ADEKKFWGK.Y
157 - 160	537.2820	536.2747	536.2747	0.0000 0	K.FWGK.Y
157 - 168	1601.8587	1600.8514	1600.8514	0.0000 2	K.FWGKLYEIIARR.H
169 - 183	1888.9268	1887.9195	1887.9195	-0.0000 0	R.HPYFYAPELLYANK.Y
184 - 209	3134.4986	3133.4913	3132.4137	1.0776 2	K.YNGVFQEQCAEDKGA ^{.....} CLLPKIETMR.E + Oxidation (M)
198 - 209	1388.5708	1387.5635	1387.7316	-0.1680 1	K.GACLLPKIETMR.E
205 - 218	1590.8632	1589.8559	1589.8559	0.0000 2	K.IETMREKVLASSAR.Q
219 - 222	572.3627	571.3554	571.3554	0.0000 1	R.QRLR.C
229 - 235	820.4675	819.4602	819.4603	-0.0000 1	K.FGERALK.A
236 - 245	1145.6425	1144.6352	1144.6353	-0.0000 1	K.AWSVARLSQK.F
242 - 248	847.5036	846.4963	846.4963	0.0000 1	R.LSQKFPK.A
246 - 263	2065.1579	2064.1506	2064.1507	-0.0001 2	K.FPKAEFVEVTKLVTDLTK.V
257 - 263	789.4716	788.4643	788.4644	-0.0000 0	K.LVTDLTK.V
257 - 266	1153.6939	1152.6866	1152.6867	-0.0000 1	K.LVTDLTKVHK.E
281 - 285	517.2980	516.2907	516.2907	-0.0000 0	R.ADLAK.Y
341 - 359	2301.0822	2300.0749	2300.0749	0.0000 1	K.NYQEAKDAPLGSFLYEYSR.R
347 - 371	2988.5366	2987.5293	2987.5293	-0.0000 2	K.DAPLGSFLYEYSRRHPEYAVSVLLR.L
360 - 371	1439.8117	1438.8044	1438.8045	-0.0000 1	R.RHPEYAVSVLLR.L
361 - 374	1595.9267	1594.9194	1594.9195	-0.0001 1	R.HPEYAVSVLLRLAK.E
413 - 433	2530.1006	2529.0933	2528.2118	0.8815 1	K.QNCDQFEKLGEGYGFQNALIVR.Y
434 - 437	567.3249	566.3176	566.3176	-0.0000 1	R.YTRK.V
452 - 459	817.4890	816.4817	816.4818	-0.0000 1	R.SLGKVGTR.C
456 - 468	1578.5981	1577.5908	1578.7243	-1.1334 1	K.VGTRCCTKPESER.M
499 - 507	1138.5673	1137.5600	1137.4907	0.0693 0	K.CCTESLVNR.R
545 - 548	516.3504	515.3431	515.3431	0.0000 1	K.QIKK.Q
548 - 557	1142.7143	1141.7070	1141.7070	-0.0000 1	K.KQTALVELLK.H
549 - 568	2304.3285	2303.3212	2303.3212	-0.0000 2	K.QTALVELLKHKPKATEEQLK.T
558 - 580	2689.4017	2688.3944	2688.3945	-0.0001 2	K.HKPKATEEQLKTVMENFVAFVDK.C
562 - 568	818.4254	817.4181	817.4181	0.0000 0	K.ATEEQLK.T
562 - 580	2199.1001	2198.0928	2198.0929	-0.0001 1	K.ATEEQLKTVMENFVAFVDK.C
569 - 580	1399.6926	1398.6853	1398.6853	-0.0000 0	K.TVMENFVAFVDK.C
569 - 587	2220.1369	2219.1296	2218.9697	0.1599 1	K.TVMENFVAFVDKCCAADDK.E
588 - 607	2091.0426	2090.0353	2090.0718	-0.0365 1	K.EACFAVEGPKLVVSTQTALA.-

(d)

Figure 11.12 (Continued)

Peptide Spectral Matching

The most common method of protein identification from biological mixtures, which involves peptide sequence inference from shotgun LC-MS/MS datasets, is usually done by a database search approach or through peptide spectral matching (PSM), where the acquired MS² spectra in one of the several MS data formats (see Reporting Standards) are searched against a compiled set of annotated protein sequences obtained from a curated public database such as UniProt or NCBI nr (see Chapter 1; Table 11.1). In all database search algorithms, each entry in the database is first digested *in silico* by applying the same specificity rules as for the enzyme actually used to digest the experimental sample. Then, each experimental MS² spectrum is correlated against theoretical fragmentation patterns constructed for each peptide using common fragmentation rules that consider ions for amino acids with the same (i.e. isobaric) mass, the loss of ammonia and water ions, and the spectral intensity of ions, to find an appropriate match. The search is typically restricted to a subset of peptides that meet criteria set by the user,

Table 11.1 List of common sources of protein sequences (used in FASTA format).

Database	Type	URL
UniProt	Reference proteomes	www.uniprot.org/proteomes
NCBI – Protein	Reference proteomes	www.ncbi.nlm.nih.gov/protein
Ensembl	Reference proteomes	www.ensembl.org/info/data/ftp/index.html
PATRIC	Reference proteomes	www.patricbrc.org
WormBase	Nematode genomes	www.wormbase.org
FlyBase	Drosophila genomes	flybase.org

such as mass tolerance, proteolytic enzyme constraints, and allowance for missed cleavages or the presence of a possible PTM.

The output of the search is a list of candidate matches (both peptide sequences and cognate proteins) that are assigned a score and ranked to define the best possible candidate. Different database search tools use different scoring schemes to compute a likelihood score for each match to discriminate between potentially correct and likely incorrect assignments. Several effective MS^2 database search tools are currently available, including established and widely used commercially distributed applications such as SEQUEST and Mascot, as well as freely available ones such as X! Tandem, Andromeda/MaxQuant, and MS-GF+ (see Internet Resources).

To make peptide identifications as reliable as possible, most algorithms also search the query MS^2 spectra against randomized or reversed decoy versions of the same reference sequence to define and minimize the FDR (i.e. to calculate the number of random matches for a given score as a function of non-random matches). Unreliable identifications are then filtered from the results by setting a stringent scoring threshold that minimizes false positives while retaining reasonable putative identifications. The finalized list of identified peptides is then assembled into cognate proteins (protein inference) through data normalization and statistical assessment following the database searching.

The identification of PTMs using MS^2 is even more computationally intensive and error prone, as it involves searching all potential combinations of mass shifts across most peptide sequences in the protein database. This results in a combinatorial explosion in the number of potential candidates to be matched. Hence, database search tools only recommend searching for up to, at most, two or three different modifications in a single run. Most conventional database search tools like Mascot, SEQUEST, and MaxQuant are limited to the detection of a fixed number of pre-specified PTMs. However, more flexible algorithms employing a “blind” or PTM-agnostic search strategies, like Sequential Interval Motif Search (SIMS), or through hybrid search approaches like those implemented in GutenTag, InsPecT, and PEAKS PTM, have been devised to identify unspecified PTMs. As the search space is largely unbounded, as a pragmatic constraint, hybrid searches generate an initial error-tolerant de novo search by which to narrow down potential candidate sequences, or even a first pass conventional database search to filter down a smaller protein pool.

De Novo Peptide Sequencing

The standard sequence database search approach fails to identify novel peptides that are not represented in the reference repository used and also cannot be used in cases where the corresponding genome sequence of the organism under question is unavailable or incomplete. In such circumstances, de novo sequencing is an alternative approach where a peptide spectrum is sequenced without prior knowledge of extant amino acid sequences.

De novo sequencing uses the sequential mass differences between two adjacent fragment ions to cumulatively calculate the mass of the corresponding amino acid residues present in

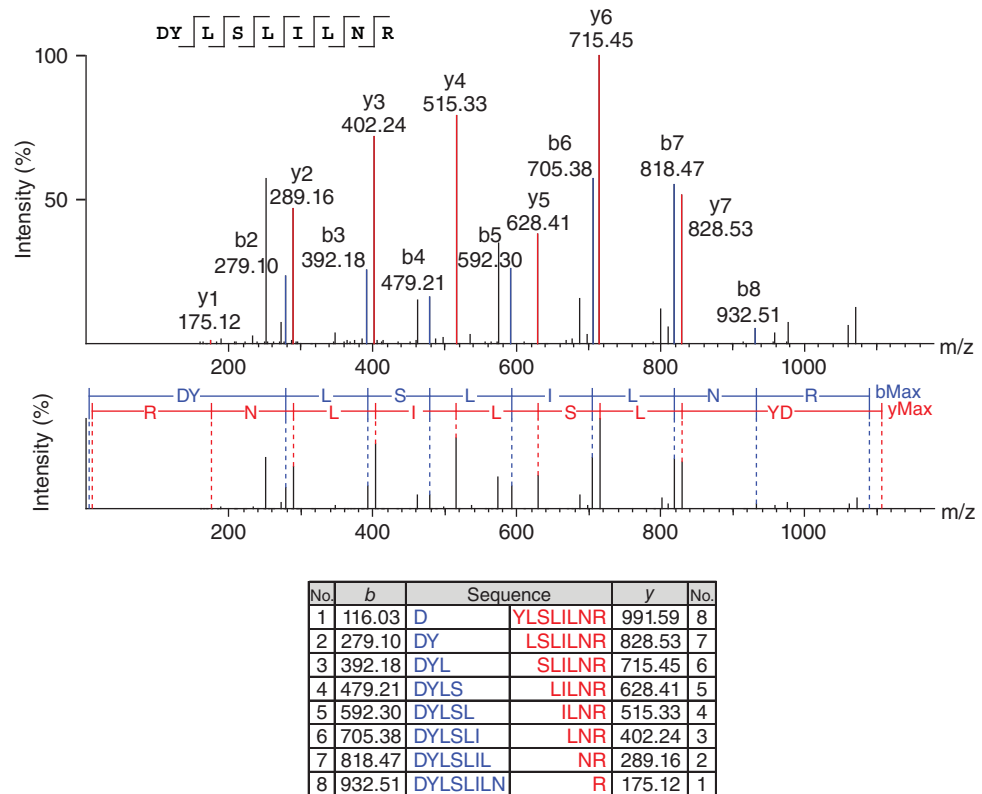


Figure 11.13 Peptide sequencing via tandem mass spectrometry (MS/MS) spectra interpretation. Annotated MS² spectrum showing peptide fragment peaks representing *b*- and *y*-ions. De novo sequencing algorithms use the mass difference between proximal fragment ion pairs to calculate the mass of the corresponding cognate amino acid residue in an iterative process to determine the sequence along the peptide backbone. For example, the mass difference between the *y*₅ and *y*₆ ions is equal to 87.04 Da, which corresponds to the exact mass of serine (S). Similarly, the next residue between *y*₅ and *y*₄ can be determined as leucine (or the isobaric residue isoleucine) based on the corresponding mass difference. Screen shots showing the PEAKS de novo search engine, highlighting an annotated spectrum and derived sequence of a candidate peptide.

the peptide backbone. Identification of discrete peak ion types is a crucial feature of the de novo search algorithm (Figure 11.13). For example, using either the “*b*” or “*y*” ion series produced by collision-induced dissociation (CID) fragmentation, a set of amino acid sequences is generated that is also consistent with the measured mass of the intact peptide (Box 11.1). Based on a variety of criteria, such as spectral deconvolution and filtration of homeometric peptides (different peptides with similar theoretical identical sets of *b*- and *y*-peaks), candidate sequences (often in the tens of thousands) are narrowed down to best fit to the experimental MS² spectrum. The advantages of the de novo approach are that it is not affected by sequence errors in the search database and can use partial sequences to search for PTMs. However, as with blind PTM searching, it is a computationally intensive and error-prone process, and so is especially dependent on high-quality MS² spectra that are complete, of high accuracy, and free of spurious noise. Some examples of popular de novo software tools include Lutefisk, PEAKS, and PepNovo+ (see Internet resources).

Spectral Library Searching

Spectral library searching has emerged as an alternative method to traditional protein sequence database searching, particularly for DDA-based data generation procedures. In theory, for a given a sequence, a library MS² spectrum represents the fragment intensities and

types of observed fragment ions with greater fidelity than an *in silico*-predicted MS² spectrum generated computationally by a database search for the same peptide sequence. Once an MS² peptide spectrum has been confidently matched, using a traditional shotgun sequencing proteomics pipeline, it is stored in an annotated spectral library (ASL) and can later be re-used for rapid identification of additional MS² spectra produced by the same peptide in another experiment. As this approach does not rely on access to conventional protein sequences, with a vast number of unverified candidates, it is extremely fast when compared with the traditional database search methods. A pairwise spectral comparison takes milliseconds to perform, as opposed to the minutes taken by standard database matching methods, hence providing a more efficient and potentially more reliable way to identify MS² spectra.

A spectral library search is essentially a pattern matching strategy that has been used in analytical spectroscopy since the 1950s. However, its use as a proteomics search tool has only become possible in the last two decades owing to the availability of proteome-wide MS² spectra of representative specimens that have made the construction of representative ASLs possible. Spectral library searching is fast becoming an ideal tool in applications such as instrumentation quality control, molecular scanners, and biomarker validation, where obtaining a speedy and confident match to a pre-defined target is of the utmost importance.

The National Institute of Standards and Technology (NIST; Stein 1990) and Global Proteome Machine (GPM; Fenyö et al. 2010) databases are two publicly available reference peptide spectral libraries collectively containing over 6 billion annotated spectra from 16 million distinct peptides. They are constantly being updated as more and more high-quality MS² data become available. Since the goal is rapid identification, a spectral library search engine simply requires an annotated MS² spectral library along with defined rules specifying the protease used in protein digestion. The list of candidate peptide-spectrum matches obtained is first filtered by aligning the precursor mass and then scored based on the calculated Pearson correlation with the experimental MS² spectrum. A match score is computed to capture the similarity between the experimental and library MS² spectrum. Candidates are ranked based on the scores, and the highest scoring peptide from the library is assigned to the spectrum. Since the spectral library is derived from experimentally observed MS² spectra, this schema imparts a higher identification sensitivity to library searches than traditional database searches. However, one should always be aware of issues arising from peptide over- or under-representation in the spectral library.

A spectral library search can even identify peptides with unexpected PTMs that are not detectable when querying a traditional database that requires upfront knowledge of all PTMs present in the sample. This improved efficiency and sensitivity have also led to the development of spectral libraries that are specialized for the identification of PTMs. MS PepSearch from NIST, SpectraST from PeptideAtlas, and X! Hunter from GPM are some of the spectral library search algorithms in use today.

Hybrid Search

A hybrid search is an approach that combines elements of *de novo* sequencing and database sequence search approaches. In hybrid search, short peptide sequence tags (PSTs) (three to five amino acid residues long) obtained from MS² spectra are subjected to an error-tolerant database search – that is, a search that allows one or more mismatches between the sequence of the peptide that produced the MS² spectrum and the database sequence. A PST is a short amino acid sequence with prefix and suffix mass values designating its starting and ending positions in the whole peptide (Figure 11.14). In peptide sequence tagging, runs of amino acids are extrapolated from the spacing of the fragmentation peaks and these “peptide-words” are then used to identify proteins in a sequence database. This tagging technique limits the search space down to peptides in the database that contain the sequence tag, resulting in a significant reduction in search time. Representative PST search algorithms are GutenTag and InsPecT.

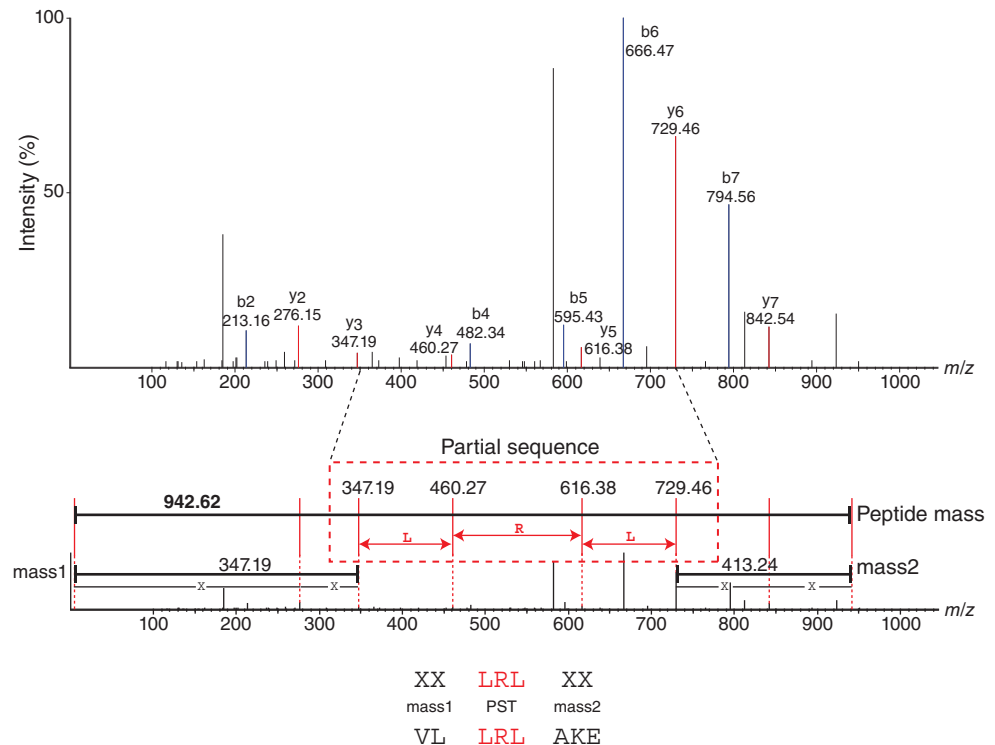


Figure 11.14 Peptide sequence tag searching. Schematic illustrating how a sequence tag (PST) or “word” is used to identify an unknown peptide from an MS² spectrum. For protein identification, the PST (LRL in the example provided) is combined with complementary information on the mass of peptide fragments before (N-terminal) the sequence tag (mass1) and after (C-terminal) the sequence tag (mass2).

Top-Down (Intact Protein) MS

Early top-down studies were hampered by restrictions on sample heterogeneity and protein sizes. However, current advances in analytical separations, such as nanoflow reverse phase liquid chromatography (RPLC), gel-eluted liquid fraction entrapment (GELFrEE), hydrophobic liquid interaction chromatography (HLIC), capillary electrophoresis (CE), and isoelectric focusing (IEF), coupled with the increasing resolution of MS instrumentation and improved ion fragmentation based on photon and electron capture methods, such as surface-induced dissociation (SID) and ultraviolet photodissociation (UVPD), have made intact protein characterization from increasingly complex mixtures feasible. Recent studies have shown the utility of the top-down approach for decoding the components of multi-proteoform-containing macromolecular complexes. Studies have even established the use of TDP in monitoring proteoforms in clinical samples, such as cerebrospinal fluid for biomarkers in pediatric brain tumor prognosis and saliva for biomarkers of early-onset Alzheimer disease in patients with Down syndrome. Some open source top-down analysis tools in use today are ProSight PTM, TopPIC, MS-Align+, and the recently released Informed-Proteomics.

Database Search Models

The massive amounts of MS² spectra generated by modern proteomics platforms (often millions of spectra per study) can only be analyzed by the use of automated search engines or software platforms. Numerous scoring algorithms have been devised, but those in use today can be separated into two general classes: those that require interpretation for selection of specific ion mass features based on the ion peaks present in a spectrum prior to sequence searching and scoring, and those that require no interpretation and that attempt to score all available mass peaks. De novo sequencing algorithms (that infer polypeptide sequences from scratch)

belong to the first class, while standard database search algorithms (that perform sequence matching) follow the uninterpreted approach. Both types of algorithms have advantages and disadvantages, but either type can be used to effectively identify polypeptide sequences from high-quality MS² spectra.

Uninterpreted search algorithms can be further subclassified into four different subtypes on the basis of the scoring approach used for PSM: descriptive, interpretative, stochastic, and probability-based matching. Descriptive algorithms are based on matching the experimental to the theoretical spectrum through correlation analysis, while interpretative models try to interpret a partial sequence from an MS² spectrum prior to a database search. Probability-based matching models derive the probability of the peptide identification by establishing a statistical relationship between the experimental spectrum and the database, and the stochastic scoring models make use of training sets of known spectra to derive the probability of the best match.

PSM Software

The aim of all search engines is to decipher an MS² spectrum obtained from the fragmentation of a peptide by selecting a list of best matching candidates using various scoring schemes to define a match and for assembling multiple identified peptides into their associated proteins. While search engines using a probabilistic scoring approach attempt to discriminate between true and false identifications, non-statistically scoring search engines depend on the subsequent application of a statistical tool such as PeptideProphet, StatQuest (Kislinger et al. 2003), or Percolator to convert the initial matching score into a likelihood or probability. Some search engines can be used as stand-alone applications on the identification and even quantification for data obtained using specialized MS techniques, while others are integrated into large software packages or platforms that allow for more complete and user-friendly MS data analysis. Some tools (such as MaxQuant and MS-GF+) are open access and freely available, while others are proprietary commercial packages that need to be licensed for use. Certain tools offer online versions that allow restricted analysis, with the latest trend being the use of cloud computing services such as Amazon Web Services (Halligan et al. 2009) and ProteoCloud (Muth et al. 2013). However, most applications need to be run on a local computer or cluster with sufficient computing power. While an in-depth discussion of all the key attributes of the many currently available search tools is beyond the scope of this chapter, some aspects of the more widely used tools are covered briefly below.

SEQUEST

The SEQUEST search algorithm is a robust descriptive scoring approach introduced by Eng, Yates, and colleagues at the University of Washington (Eng et al. 1994). It was the first and now one of the most widely used automated database search tools devised for peptide identification from MS² data. The SEQUEST algorithm pre-processes MS data through an iterative peptide-spectrum matching strategy based on the precursor mass and a user-specified tolerance, followed by peak binning and normalization. The pre-processed data are then scored using a two-step scoring approach, wherein a preliminary score (S_p) is first calculated based on the number of ions in the MS² spectrum that match with the experimental data. Theoretically constructed spectra are then generated for the top-ranked 500 candidate peptides and systematically evaluated against the experimental spectrum to generate a normalized cross-correlation score (X_{corr}), which is a scalar dot product with a correction factor (Figure 11.15). The peptide with the highest X_{corr} value is deemed the best match, with match quality and uniqueness further judged based on the difference between the top and next best match by calculation of a Delta correlation (ΔCn) score. This cross-correlation analysis is the primary function implemented within SEQUEST and makes the tool particularly sensitive albeit computationally intensive (i.e. slow).

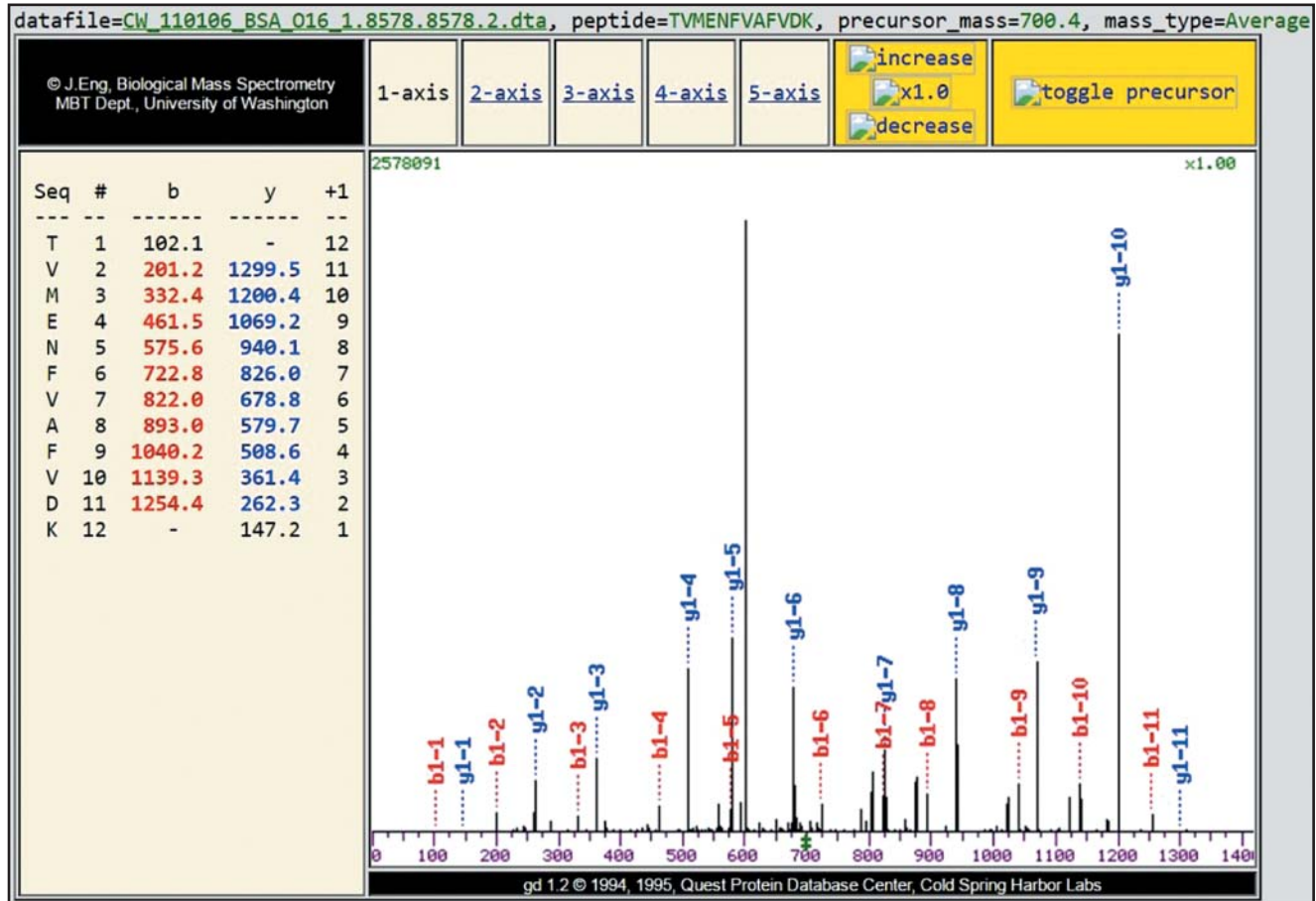


Figure 11.15 Peptide spectrum match (PSM). Annotated MS² spectrum showing matched ion series for a representative BSA (bovine serum albumin) peptide identified using the SEQUEST search algorithm.

The ability to identify dynamic (variable) modifications was added in subsequent updates, and SEQUEST was later integrated into the commercial Proteome Discoverer software suite. Efforts to develop faster versions of SEQUEST were subsequently introduced by adding a pre-computed indexing function to speed up the calculation of X_{Corr} . TurboSEQUEST was developed as part of the Crux software suite, followed by a much faster implementation called Tide. By means of algorithmic enhancements and better use of limiting computer resources, Tide is geared toward high disk usage and can be used for parallel execution on a CPU cluster by running multiple program instances. It is freely available for academic and non-profit use as a part of the Crux software toolkit.

X! Tandem

X! Tandem is an open source search engine within the X! suite of database matching algorithms that is distributed as part of the GPM. It was originally implemented as a collaborative effort between Robertson Craig and Ronald Beavis as a free open source search tool called TANDEM (Craig and Beavis 2004), in contrast to most popular search tools that were proprietary and offered limited scope for further improvement. The implementation of TANDEM was undertaken with a view to optimizing speed and improving identifications, and was designed to run from the command line. It pre-processes experimentally derived spectra to remove spurious peaks (noise) and to generate enzyme-specific theoretical spectra from

a protein sequence while considering potential post-translational and chemical modifications when matching the observed and predicted spectra. A hyperscore based on the hypergeometric distribution is then calculated as the dot product of the sum of matched peak intensities and the factorial of the number of matched *b* and *y* ions.

Two additional scoring metrics – the *K*-score and *S*-score – have been introduced to measure the similarity between peptide MS² spectra and sequence candidates. Similar to the native hyperscore, the *K*-score includes a pre-processing step that takes input from noised and unmatched peaks to give a more sensitive match, while the *S*-score divides the sum of log intensities of matched peaks by the square root of the sequence length, with statistical significance assessed by an expectation (*e*) value. Once the peptide evidence is established, the protein-level inference is estimated using a Bayesian model based on the number of peptides identified for a given protein and their respective scores.

TANDEM was one of the first programs to use Biopolymer Markup Language (BIOML; Fenyő 1999), an Extensible Markup Language (XML) format designed for annotation of protein sequence information and generating input and output files according to the standard reporting formats of analytical instruments, allowing for easy integration into MS search pipelines. TANDEM has been integrated into the Trans-Proteomic Pipeline (TPP) software suite, one of the more popular public MS² analysis platforms, but is still maintained as part of the X! suite of tools by GPM with the latest version at the time of this writing being Alanine (2017.02.01).

MaxQuant (Andromeda)

Andromeda is a database search engine developed for the powerful MaxQuant software suite (Cox et al. 2011) that implements a probabilistic scoring algorithm for PSM scoring. It is capable of handling MS² spectra generated with high fragment mass accuracy and can assign, score, and quantify complex patterns of PTMs such as multiply phosphorylated peptides, while maintaining the ability to search efficiently across large sequence databases.

As with most search engines, a user specifies allowed peptide and protein modifications, the enzyme(s) used for protein cleavage, and the reference protein sequence database to be searched. A list of all peptides in the database is then generated using these parameters and indexed using a two-layer structure based on peptide mass for fast retrieval. Based on the peptide sequence and configuration of fixed and variable modifications for a given peptide, the theoretical fragment ions are calculated after averaging and deconvolution of multiple charge states. The Andromeda scoring function is based on the binomial distribution probability formula. The MS² spectrum is divided into mass ranges of 100 Th (mass-to-charge ratio units); the score calculated as 10 times the logarithm of the probability of the number of matches between the experimental peaks and theoretical fragment masses across the spectrum range while taking into account peptide length, number of missed cleavages, and potential presence of modifications. Peptide identifications are then filtered using a statistically determined cut-off based on a target–decoy-derived FDR and mapped to cognate proteins.

Andromeda was developed with a robust architecture and unlimited scalability. It can function independently or as a search engine integrated within the MaxQuant computational platform, using a graphical user interface that was specifically developed for high-resolution (Orbitrap) MS data. This includes peak detection in the raw data, quantification, scoring of peptides, reporting of protein groups, and support of both quantitative label-free (e.g. spectral counts) and isotopic (e.g. stable isotope labeling of amino acids in a cell culture, SILAC) and isobaric (e.g. tandem mass tag, TMT) labeling techniques. Both tools are freely available (see Internet Resources) and can be run on a Windows desktop computer, eliminating client–server setup and network communication issues. For searching individual spectra, Andromeda is also accessible via a web server and can be run from the command line. To assist biological interpretation, further downstream biological analysis of MaxQuant/Andromeda results can be done using a separate module called Perseus.

PSM on the Web

A restricted web accessible version of Mascot for MS² is available for single sample searches. It is very similar to the PMF search with some modifications. Here the data are submitted online in the form of peak lists obtained from converting raw data through a process known as peak picking (Figure 11.8a). Each peak list is made up of the observed peptide ion mass values and, optionally, associated intensity values where available. It can also be submitted as a Mascot generic format or .mgf file. The current version of Mascot also supports vendor-specific formats such as .dta (SEQUEST), .asc (Finnigan), .pkl (Micromass), as well as standard formats adopted by the proteomics community such as .mzML and .mzData (see Reporting Standards). Apart from the regular search-specific parameters (see PMF on the Web), other additional parameters that can be set include MS² or fragment ion error tolerance and the quantitation method for labeled or label-free samples. One can also select the type of MS instrument and ion activation method used from the list provided, the charge state of peptide fragments, and whether to run a decoy search to calculate the FDR. At the end of a search, a detailed summary report is generated encapsulating the putative peptide and protein identifications (Figure 11.16). In its current iteration, Mascot supports PMF, PST, and standard database searching of MS² spectra, along with PTM identification and relative quantification using label and label-free techniques. A free (but restricted) version for all three search types is available online (see Internet Resources); high-throughput operation is accessible commercially.

Reporting Standards

One of the pivotal elements for development and progress in any field of research is the need for collaboration and easy exchange of data. For that to happen, it becomes crucial that MS data adhere to a common standard to allow interoperability between software tools and computing platforms, as well as for deposition of proteomics data in public repositories to facilitate sharing, reuse and, ultimately, new biomedical insights leading to clinical translation. In order to establish MS data standards, the Human Proteome Organization (HUPO) formed the Proteomics Standard Initiative (PSI) in 2002 (Orchard et al. 2003). The goal of this effort is to develop community standard reporting formats using minimum information guidelines and controlled vocabularies; it also promotes the development of public resources and tools for data distribution through group charters addressing different aspects of MS-based proteomics. These include PSI-MI (Proteomics Standard Initiative – Molecular Interactions), a data format for reporting and exchange of molecular interactions (Chapter 13), MIAPE-MS (Minimum Information About a Proteomics Experiment – Mass Spectrometry) for experimental data, MIAPE-MSI (Minimum Information About a Proteomics Experiment – Mass Spectrometry Informatics) for MS data analysis, MIASSPE (Minimum Information About Sample Preparation for a Phosphoproteomics Experiment) for PTMs such as phosphoproteomics, and MIAPE-Quant (Minimum Information About a Proteomics Experiment – Mass Spectrometry Quantification) for proteomics quantification experiments. These guidelines define the basic data elements and metadata required for MS data release, while the data formats provide models for reporting the information to be shared. The latter include ad hoc formats that represent the needs of a specific group or developer. In addition to ad hoc formats, there are de facto standards, such as pepXML and protXML developed as a part of the TPP suite, that have not been through a standardization process but are nevertheless commonly accepted. Actual standards, such as mzML, are defined through a formal standardization process, which defines the structure of the XML format after extensive testing and review.

Proteomics XML Formats

Many proteomics data formats exist. Unfortunately, vendor-specific proprietary MS data formats do not allow data to be easily manipulated or shared. To overcome these problems, several

MATRIX SCIENCE Search this site

Home Mascot database search Products Technical support Training News Blog Newsletter Contact

Access Mascot Server | Database search help

Mascot database search > Access Mascot Server > MS/MS Ions Search

MASCOT MS/MS Ions Search

Your name: abcd Email: abcd@xyz.com

Search title: MB_Cyt_60

Database(s): contaminants (AA), SwissProt (AA)

Amino acid (AA): cRAP, NCBIProt

Nucleic acid (NA): Environmental_EST, Fungi_EST, Human_EST, Invertebrates_EST, Mammals_EST, Mus_EST

Taxonomy: Mus musculus (house mouse)

Enzyme: Trypsin Allow up to: 1 missed cleavages

Quantitation: Label-free [MD]

Fixed modifications: Carbamidomethyl (C)

Variable modifications: Oxidation (M)

Acetyl (K), Acetyl (N-term), Acetyl (Protein N-term), Amidated (C-term), Amidated (Protein C-term), Ammonia-loss (N-term C), Biotin (K), Biotin (N-term), Carbamyl (K), Carbamyl (N-term), Carboxymethyl (C)

Peptide tol. ±: 1.2 Da # ¹³C: 0 MS/MS tol. ±: 0.6 Da

Peptide charge: 2+ Monoisotopic: Average:

Data file: Choose File merge10.mgf

Data format: Mascot generic Precursor: m/z

Instrument: ETD-TRAP Error tolerant:

Decoy: Report top: AUTO hits

Start Search ... Reset Form

(a)

Figure 11.16 Mascot search engine. Mascot MS² database search submission window and representative peptide spectrum match (PSM) search results. (a) Tandem mass spectrometry (MS/MS) ion search submission form where search-specific parameters such as an enzyme, number of missed cleavages, organism taxonomy modifications, quantitation, precursor m/z , MS instrument, and ion activation (fragmentation) mechanism can be set or selected. (b) The search results page displays a ranked list of proteins each with the $-\log_{10}(P)$ protein score. The protein with the highest significance score is considered the most likely match. Clicking on the protein name brings up the corresponding peptide view for that protein. (c) The peptide view page displays a scrollable panel for viewing the mass spectrum of each peptide identified for the protein along with the fragmentation table listing the masses of the peaks in the mass spectrum. A scored list of peptides identified for that protein is also displayed.

MATRIX SCIENCE **MASCOT Search Results**

User : abcd
E-mail : abcd@xyz.com
Search title : MB_Cyt_60
MS data file : merge10.mgf
Databases : 1: contaminants 20090624 (262 sequences; 133,770 residues)
 2: SwissProt 2017_05 (554,515 sequences; 198,509,421 residues)
Taxonomy : 1: (none)
 2: Mus musculus (house mouse) (16,884 sequences)
Timestamp : 7 Jun 2017 at 16:15:17 GMT

Not what you expected? Try [the select summary](#).

▶ **Search parameters**
 ▶ **Score distribution**
 ▶ **Modification statistics**
 ▶ **Legend**

Protein Family Summary

Significance threshold p< 0.05 Max. number of families AUTO
 Display non-sig. matches Dendrograms cut at 0
 Show Percolator scores
 Preferred taxonomy All entries ▼

▶ **Sensitivity and FDR (reversed protein sequences)**

Protein families 1–12 (out of 12)

20 ▼ per page 1

▼ **1** **2::UBE3C_MOUSE** 35 Ubiquitin-protein ligase E3C OS=Mus musculus GN=Ube3c PE=1 SV=2

	Score	Mass	Matches	Sequences	emPAI	
1.1	2::UBE3C_MOUSE	35	125037	1 (1)	1 (1)	0.03

Ubiquitin-protein ligase E3C OS=Mus musculus GN=Ube3c PE=1 SV=2

▼ **1 peptide matches (1 non-duplicate, 0 duplicate)**

Query Dupes	Observed	Mr (expt)	Mr (calc)	Delta M	Score	Expect	Rank	U	Peptide
548	747.0494	746.0422	745.4082	0.6339	0	35	0.013	1	K.VSLGGASR.K

▼ **2** **2::FGFP3_MOUSE** 30 Fibroblast growth factor-binding protein 3 OS=Mus musculus GN=Fgfbp3 PE=1 SV=2

	Score	Mass	Matches	Sequences	emPAI	
2.1	2::FGFP3_MOUSE	30	26714	1 (1)	1 (1)	0.12

Fibroblast growth factor-binding protein 3 OS=Mus musculus GN=Fgfbp3 PE=1 SV=2

▼ **1 peptide matches (1 non-duplicate, 0 duplicate)**

Query Dupes	Observed	Mr (expt)	Mr (calc)	Delta M	Score	Expect	Rank	U	Peptide
452	674.8212	673.8139	673.3507	0.4632	1	30	0.0086	1	R.DKGAAGR.E

▶ **3** **2::DDIAS_MOUSE** 28 DNA damage-induced apoptosis suppressor protein OS=Mus musculus GN=Ddias PE=2 SV=1

▶ **4** **2::SRFB1_MOUSE** 21 Serum response factor-binding protein 1 OS=Mus musculus GN=Srfbp1 PE=1 SV=1

▶ **5** **2::WASF1_MOUSE** 19 Wiskott-Aldrich syndrome protein family member 1 OS=Mus musculus GN=Wasf1 PE=1 SV=2

▶ **6** **2::HEPH_MOUSE** 18 Hephaestin OS=Mus musculus GN=Heph PE=1 SV=3

▶ **7** **2::CL004_MOUSE** 18 Protein Cl2orf4 homolog OS=Mus musculus GN=D6Wsu163e PE=1 SV=1

▶ **8** **2::CLC2L_MOUSE** 18 C-type lectin domain family 2 member L OS=Mus musculus GN=Clec2l PE=1 SV=1

▶ **9** **2::CATC_MOUSE** 15 Dipeptidyl peptidase 1 OS=Mus musculus GN=Ctsc PE=1 SV=1

▶ **10** **2::GALR3_MOUSE** 15 Galanin receptor type 3 OS=Mus musculus GN=Galr3 PE=2 SV=2

▶ **11** **2::SUCC_MOUSE** 15 SUN domain-containing ossification factor OS=Mus musculus GN=Suco PE=1 SV=3

▶ **12** **2::MEIOC_MOUSE** 14 Meiosis-specific coiled-coil domain-containing protein MEIOC OS=Mus musculus GN=Meloc PE=1 SV=1

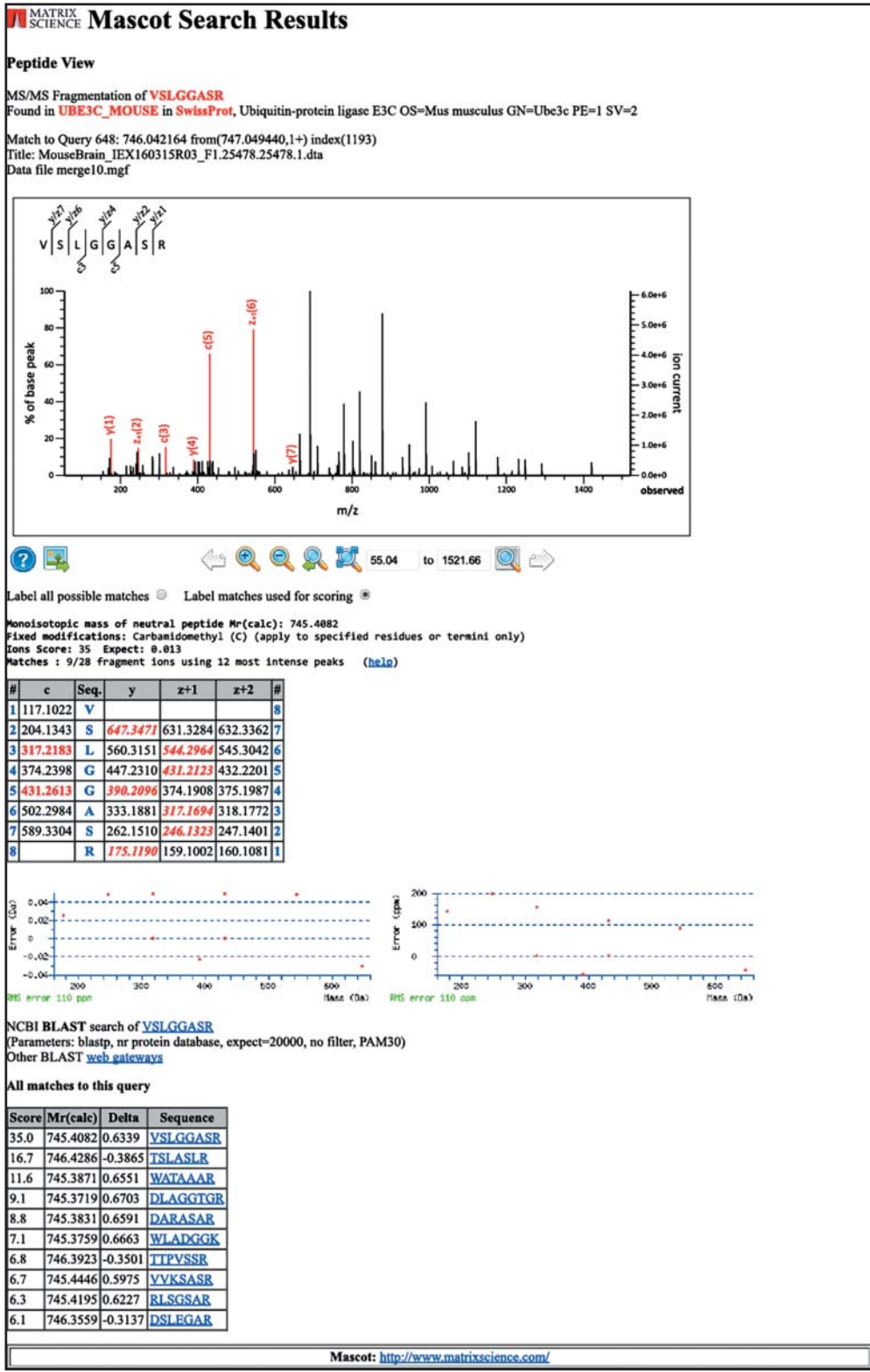
20 ▼ per page 1

Not what you expected? Try [the select summary](#).

Mascot: <http://www.matrixscience.com/>

(b)

Figure 11.16 (Continued)



(c)

Figure 11.16 (Continued)

open data formats have emerged over the past decade. The latest recommended open standard proteomics format is mzML, developed and supported by HUPO PSI and building upon earlier open mzData and mzXML standards; these standards were initially widely used to store raw MS data such as spectra and chromatograms. While mzML is the best standard currently available, older formats, such as the mzXML format developed by the Institute for Systems Biology (ISB, Seattle, Washington), are still widely used. These standards are written in XML and contain a textual representation of proteomics data structures with emphasis on simplicity and usability that make them both human and machine readable.

Since its inception, PSI has defined other data formats such as TraML for devising transition lists as input to target-directed SRM experiments, mzIdentML for peptide and protein identifications, mzQuantML for quantitative MS data, mzTab for proteomics and metabolomics results, gelML for protein separation methods, and spML for sample processing. The definition and availability of these PSI data formats have helped to streamline the development of MS algorithms and software platforms leading to increased interoperability and data exchange. Examples of the many tools capable of standard-compliant implementation of PSI formats in use today are ProteoWizard, PRIDE, and OpenMS.

Proteomics Data Repositories

The endpoint of most proteomics projects is reached when a manuscript is published. In the early days of proteomics (up until the mid-2000s), standard practice was to release the final processed data as supplemental information at the time of publication. As a result, there was no way for the community to access the unprocessed data or raw experimental results unless requested of the authors. Even that could be problematic if the data were not archived properly and could not be traced. In the mid-2000s, many journals began requesting proteomics data deposition into a public repository coincident with the publication, similar to the established practice in the DNA sequencing field. With steady advances in MS data generation, demand for access to raw proteomics data has increased manifold, resulting in mandatory deposition of enormous amounts of experimental data into public repositories. Access to data allows re-use and re-analysis by other researchers, allowing the definition of MS-observable proteomes or annotated spectral libraries.

Leading data repositories for proteomics data include PeptideAtlas, PRIDE, GPMdb, the Mass Spectrometry Interactive Virtual Environment (MassIVE), jPOST, iProX, Chorus, and the PeptideAtlas SRM Experiment Library (PASSEL). The ProteomeXchange (PX), representing a consortium of repositories, was developed to manage the integration of public repositories and data sharing with the scientific community.

ProteomeXchange

The PX Consortium (Deutsch et al. 2017) was launched to oversee standardization of submission guidelines for proteomics MS data. PX provides a user-friendly data deposition procedure and framework to coordinate the resources of existing database repositories, including PRIDE, MassIVE, jPOST, iProX, PASSEL, and PeptideAtlas. Mandatory data and metadata required for submission are MS output (raw data in a binary format or the standard mzML format), processed identification results, and metadata that describe experimental conditions. Other information such as peak lists and quantification results can also be provided. Once submitted, authors are able to cite an assigned PX accession in their pre-publication. While the first five databases store user-submitted data and are considered a primary resource, data in PeptideAtlas are re-processed through the TPP pipeline, similar to what is done by GPMdb, and constitute a secondary resource. In addition to PeptideAtlas, PASSEL was set up as a repository for target-driven SRM data. To date, more than 4500 datasets covering over 900 organisms have been submitted, with Proteome Central serving as an access portal with browsing and advanced visualization features.

PRIDE

The Proteomics Identifications database (PRIDE; Vizcaíno et al. 2016) is a repository for MS data that includes actual spectra, as well as tentative peptide and protein identifications and PTM site assignments. Data supporting a scientific publication can be deposited in PRIDE before or during the peer review process and is assigned a PX accession number. After publication, the data are made publicly available and can be downloaded using the provided accession number. The database can be queried by PX accession, protein accession, PubMed accession, or any of the keywords included in the metadata, and the data stored in a number of formats using multiple tools: the PRIDE Converter tool can convert uploaded MS spectra and identifications to the PRIDE XML format, PRIDE Inspector is an XML validator for verifying data formatting prior to submission, the PRIDE Archive web page can be used to query the database, and PRIDE Cluster can group spectra in the repository based on similarity, with clusters queried using a peptide sequence or consensus spectrum. Additionally, species-specific spectral libraries can be downloaded.

The submission process can be in the form of a complete submission, where the processed identification data are first converted to PRIDE's XML format. PRIDE also supports partial submissions, where the PSI de facto mzXML format and standard mzML or mzIdentML are provided; the corresponding peak list file of the search engine used must also be included. Complete submission ensures that processed data are integrated into PRIDE, supporting the connection of processed results directly with the mass spectra, enabling quality assessment using the database's visualization tools.

PeptideAtlas

PeptideAtlas (Farrah et al. 2013) was primarily developed as a database for the annotation of eukaryotic peptide sequences but has expanded to serve as a framework for storage, exchange, and integration of proteomics data. PeptideAtlas re-processes high-throughput data to a stringent FDR assessment using the TPP before mapping the resultant peptide annotations to genomes (unlike PRIDE, which stores and presents peptide and protein identifications as submitted by the investigator). After data are uploaded to PeptideAtlas through its submission interface, the reprocessed data are organized into "builds" belonging to a proteome (or sub-proteome). PeptideAtlas also provides statistical validation tools, like PeptideProphet and ProteinProphet, to control false-positive identifications and is now a highly curated protein expression database. More recently, PeptideAtlas has begun to serve as a resource to build spectral libraries and SRM-related tools and is now a part of the PX consortium.

Global Proteome Machine + GPMdb

The GPM was developed with the aim of consolidating and gleaning information from burgeoning proteomics data sources for broader use in biomedical research (Craig et al. 2004). To accomplish this, the GPMdb database was set up to facilitate community access to MS² data for proteome-wide analysis using their popular open source X! suite of search tools, which includes X! Tandem and X! Hunter. Since its inception, the GPM has become a well-known protein expression database that continues to provide expansive content through the acquisition of proteomics data repositories and user submissions. The data are re-processed prior to storage for rigorous validation of peptide MS² spectra, tentative protein identifications, and PTM mappings, and are saved as ASL XML files that are indexed and stored in a MySQL database.

The GPM's X! suite of search engines allows a user to run database searches on their own data while retaining the option of submitting the results to GPMdb's annotated spectrum library. GPM also allows users to perform spectral library searches using the X! Hunter spectral library search engine and analyze data through the proteotypic peptide profiler X! P3. GPMdb categorizes the information in the database in the form of searchable interfaces,

including pYST, which provides a list of PTMs; SNAP, which provides lists of protein amino acid polymorphisms; MRM (Multiple Reaction Monitoring), which lists peptides observed in MS² experiments; and PEPTIDE, which provides species-specific peptide sequences for download. All peptides are mapped to Ensembl genome database identifiers.

Other ways of searching the GPM web interface include by accession number, peptide sequence, chromosome location, keywords, Kyoto Encyclopedia of Genes and Genomes (KEGG) pathways, gene Ontology (GO) terms, PTMs, protein amino acid polymorphisms, BRAunschweig ENzyme DAtabase (BRENDA), and tissue ontology. The GPMdb undergoes daily updates and has data covering ~829 million proteins and 8.6 billion peptides as of build 5600 (June 2019). For example, the frequency of identification of a particular protein in the GPMdb can be used as an indirect measure of confidence of how likely the protein will be identified in an MS² experiment.

Protein/Proteomics Databases

The other major type of public resource accessed by most proteomics workflows in MS spectral searching is protein sequence databases. Proteomics data closely resemble transcriptomic data (Chapter 10) or metabolomic data (Chapter 14), in that the resulting long lists of proteins (proteomics), genes (transcriptomics), or metabolites (metabolomics) need to be compared, analyzed, annotated, and biologically interpreted. The annotation and interpretation process for proteomics requires access to comprehensive protein sequence and annotation databases. These resources play a vital role in data-driven biological discovery and hypothesis generation. The vast amounts of MS spectra generated by high-throughput proteomics studies created a critical need for rigorously curated databases to aid researchers in making connections between their results and existing knowledge. Depending on the experimental design, proteomics strategies usually generate information regarding protein location, abundance, and PTMs, so functional annotation within these databases is essential in establishing the biological relevance of the identified proteins. For example, it may be possible to infer the function of a differentially regulated protein, such as its role in a biological pathway based on the functions of the interacting partners that interact or co-localize with it through data mining of functional annotation databases. In addition to well-known databases, such as the National Center for Biotechnology Information's (NCBI) RefSeq (Chapter 1), Ensembl (Chapter 4), and the Protein data Bank (PDB; Chapter 12), a few other databases relevant to proteome annotation and inference are discussed below.

UniProt

The UniProt Consortium is an authoritative and a comprehensive data store for functional information on protein sequences (Chapter 1). UniProt serves as the database of choice for protein sequences needed for MS spectral searches, as it houses proteome-level protein sequences for well-studied model organisms and other reference species with fully sequenced genomes. UniProt also contains UniRef, a database of reference sequence clusters, and UniParc, a sequence archive. Collectively, UniProt is an information-rich resource with carefully derived annotations, taxonomic information, and qualitative functional information, such as protein subcellular locations, PTMs, and pathway and disease associations, with links to available pertinent cross-references and extensive literature citations. UniProt can be used to find curated information on proteins of interest – for example, the domain structure of a protein, its biological function, subcellular location, known PTMs, role in a biological pathway, or involvement in disease as gleaned from peer-reviewed papers. It can be used to compare protein sequences to determine similar (homologous) proteins and to see associated functional information.

PTM Databases

Owing to the important role played by PTMs in regulating cellular processes and the need for a comprehensive description of PTMs by the research community, the dbPTM database was released in 2006 (Huang et al. 2019). dbPTM contains a list of all experimentally verified PTMs gleaned from public databases and putative PTMs in the UniProt database; it also offers a web-based portal for integrated access to this information, along with tools for PTM analysis. The experimentally verified subset has proved to be an excellent benchmark for evaluating the predictive power of various PTM prediction tools. This subset has also been mapped to all corresponding PDB entries to define 347 984 putative modification sites as of this writing. dbPTM also incorporates metabolic pathway information and protein–protein interactions relevant to PTM networks. The current version of dbPTM contains 908 917 non-redundant experimentally verified PTM instances, representing over 34 types of modifications, including 571 032 phosphorylation sites.

Another highly curated database for experimental mammalian PTMs is PhosphositePlus (PSP). Launched in 2003, PSP now houses over 400 000 non-redundant modification sites linked to 20 268 protein groups and 2.4 million peptides, covering 14 different modification types acquired from over 21 000 publications. While PSP incorporates data from low-throughput studies, over 95% of the PTMs are from high-throughput data, so the acquired data are re-analyzed using a common analysis standard to retain only site assignments with high probability ($p \leq 0.05$). PSP also includes structural topology and functional information about putative modification sites and provides tools for the functional analysis of PTMs with respect to protein function aspects such as disease, tissue expression, and domains.

Owing to the critical role of PTMs in cellular signaling and regulation of cellular processes, PTMs identified by proteomics studies need to be properly interpreted to gain insight into the significance of the role they may play in disease causation. Hence, databases like dbPTM and PSP serve as a valuable resource for researchers who can use them to benchmark their findings.

Selected Applications of Proteomics

The overall objective of proteomics is to study the properties of a proteome and determine the change reflected in response to various physiological states such as the cell cycle, signaling, cell division, or disease. These can be broadly classified using differential, functional, and structural proteomics strategies (Figure 11.17).

Differential Proteomics

Differential proteomics, or proteome-scale expression profiling, investigates the differences in the expression patterns of proteins between two physiological states (e.g. normal versus cancer). In biomedical research, a comparative methodology is typically used for the identification of significantly upregulated or downregulated proteins in a context- or disease-specific manner to investigate cellular responses, for use as diagnostic biomarkers, or as potential drug targets, as well as to understand the mechanistic basis of biological processes at the molecular level. Examples of differential proteomics techniques include a study identifying some of the important regulatory systems controlling glucose responsiveness in the metabolic pathway affecting diabetes (Schuit et al. 2002), the discovery of genes producing abnormal regulatory proteins in Alzheimer disease (Butterfield et al. 2003), and the identification of proteins involved in progressive dilated cardiomyopathy and heart failure (Gramolini et al. 2008).

Functional Proteomics

There are many different areas of study covered by functional proteomics, a wide-ranging term encompassing protein identification, abundance, and turnover measurements across

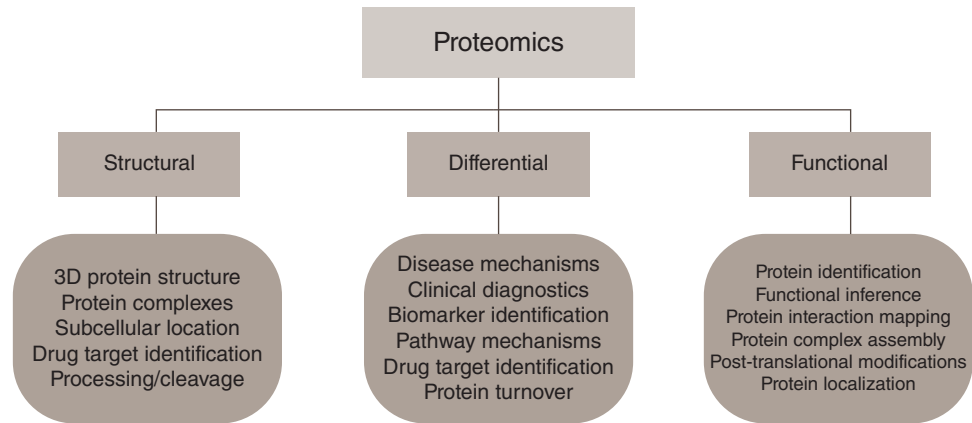


Figure 11.17 Proteomics. A broad classification of proteomics and the biological applications of proteomics studies currently and commonly being performed. 3D, three-dimensional.

changing conditions, all the way through to PTM mapping, protein localization, interaction mapping, and functional inference. For example, multi-protein complexes are known to play leading functional roles in the molecular machinery of the cell, and thus the systematic characterization of protein–protein interactions and their dynamic assembly into macromolecular assemblies is critical toward understanding their role in driving cellular signaling networks and metabolic pathways. Notably, the function of an uncharacterized subunit of a multi-protein complex physically associated with annotated components of known function can be inferred using the “guilt-by-association” (or “guilt-by-correlation”) principle (Gavin et al. 2002; Krogan et al. 2006). PTMs play an especially important role in functional proteomics, given their role in determining protein activity by impacting physical interactions such as PTM-dependent binding to proteins and nucleic acids, as well as in communicating extracellular cues via intracellular signaling cascades or driving key cellular processes as a result of protein phosphorylation/de-phosphorylation events that trigger cell division, differentiation, apoptosis, or metabolic/anabolic states.

Structural Proteomics

Proteomics studies in which the goal is to determine protein location and associations in a cell and their three-dimensional shape or structure within macromolecular complexes is called *structural proteomics*. Structural analyses can support functional characterization by providing clues about the biochemical role of a target protein through complementary information regarding biological activity and pathophysiological significance. Traditional protein biochemistry methods, while typically limited to single proteins or protein classes, can be combined with unbiased mass spectrometric techniques to study various structural aspects of protein assemblies on a growing scale (Sinz 2014).

Drug target identification is yet another application of structural proteomics, where MS is used to identify the interactions of bioactive small molecule ligands with their cellular protein targets and to define potential drug binding site(s) and three-dimensional models of a protein–ligand complex; this is a critical step toward better defining a compound’s mode of action and structure–activity relationships, as well as to assist the process of “rational drug design” and drug discovery (Djuric et al. 2016).

Summary

Similar to other areas of bioinformatics, sophisticated data analysis pipelines and algorithms are used in proteomics analysis. Careful consideration needs to be given to the quality

Table 11.2 Standard search parameters used with sequence database search engines.

	SEQUEST	!X Tandem	MaxQuant
Enzyme	Trypsin	Trypsin	Trypsin
Number of missed cleavages	2	2	2
Peptide mass tolerance	0.5 Da	0.4 Da	4.5 ppm
Maximum number of modifications per peptide	3	10	5
Fixed modifications	Carbamidomethylation	Carbamidomethylation	Carbamidomethylation
Variable modifications	Oxidation, acetylation	Oxidation, acetylation	Oxidation, acetylation
Parent mass type	Monoisotopic mass	Monoisotopic mass	Monoisotopic mass
Fragment mass type	Monoisotopic mass	Monoisotopic mass	Monoisotopic mass
Minimum peptide length	6	6	7
Maximum peptide length	40	50	25
False discovery rate	0.01	0.01	0.01
Precursor mass tolerance	10 ppm	−2.0 to 4.0 Da	6 ppm
Fragment ion method	CID	CID	CID

CID, collision-induced dissociation.

of data submitted and the parameters chosen so as to obtain optimal results. There is no “one-size-fits-all” solution that works perfectly under all circumstances, and most software tools are tailored to a specific task. The source and quality of MS data are also of the utmost importance, underlying the importance of having a thorough knowledge of the biological problem under consideration before beginning any analysis. Depending on the type of MS instrument used, the quality and type of data generated, and the kind of experimental characterization being performed, critical database search tool parameters (described in Table 11.2) need to be carefully set prior to achieving optimal performance.

Important factors that need to be considered in the context of all proteomics experiments and analyses include:

- the proper calibration of the MS instrument (e.g. using known standards)
- understanding the expected mass resolution and accuracy of the instrument
- the specification of proper proteolytic cleavage rules relevant to the protease used in protein digestion
- capturing the MS data acquisition (instrument) settings, such as:
 - ionization and fragmentation methods used, as well as the ion series identified within each spectrum
 - precursor and fragment ion mass, scan range, and match tolerance
 - presence of stable isotopes or multiple charge states
 - defining variable or pre-defined post-translational (e.g. phosphorylation) or chemical (e.g. acetylation) modifications
- the presence of contaminating species, such as trypsin autolysis products, keratin, and other experimental artifacts
- selecting the reference protein sequence database for the search
- processing and measuring the signal-to-noise ratio of each spectrum.

A good understanding of how these parameters influence the scope of the search and ultimately impact the quality of the results is of vital importance.

In general, there are two ways to ensure the quality of the results. The first is a selection of optimal parameter settings, which can be achieved by systematically varying the search parameters until a satisfactory result is obtained. For example, increasing the initial MS scan range

from 375–1500 to 400–1800 m/z can improve peptide coverage and signal to noise; broadening the search space by including orthologs from closely related but better annotated species can also provide more informative results. Another strategy to ensure high-quality search results is to integrate the results from multiple programs to achieve better coverage while minimizing false positives. Since search engines vary in their scoring schemes, taking into account different features of the input data, one algorithm may detect a feature missed by another (Kwon et al. 2011).

In all, two major considerations dictating the success of bioinformatics analysis of LC-MS/MS studies are to know the properties of the data and to be mindful that protein identification is only a first step in any proteomics analysis workflow. We trust the chapter has provided some helpful guidance in this respect.

Acknowledgments

The authors acknowledge the constructive input from members of the Emili Lab (University of Toronto, Toronto, Canada, and Boston University, Boston, MA, USA) and their assistance in compiling supporting information. We also thank Carl White and Ruth Isserlin (University of Toronto), as well as Indranil Paul and Benjamin Blum (Boston University) for sharing their expertise, sage advice, and critical insights that greatly improved the content found within this chapter.

Internet Resources

Crux	crux.ms
dbPTM	dbptm.mbc.nctu.edu.tw
Global Proteome Machine (GPM)	www.thegpm.org
GPM DB	ftp://ftp.thegpm.org/repos/peptides
GutenTAG	fields.scripps.edu/downloadfile2.php?name=GutenTag&filename=GutenTag.zip&id=3
Human Proteome Organization (HUPO)	www.hupo.org
Informed-Proteomics	github.com/PNNL-Comp-Mass-Spec/Informed-Proteomics
InsPecT	proteomics.ucsd.edu/Software/Inspect
iProX	iprox.org
jPOST	jpostdb.org
Lutefisk	www.hairyfatguy.com/lutefisk
MassIVE	massive.ucsd.edu/ProteoSAFe/static/massive.jsp
Mascot	www.matrixscience.com/cgi/search_form.pl?FORMVER=2&SEARCH=PMF
MaxQuant	www.coxdocs.org/doku.php?id=maxquant:common:download_and_installation#download_and_installation_guide
MS-Align+	bix.ucsd.edu/projects/msalign
MSblender	github.com/marcottelab/MSblender
MS-GF+	omics.pnl.gov/software/ms-gf
MS PepSearch	chemdata.nist.gov/dokuwiki/doku.php?id=peptidew:mspepsearch
OpenMS	www.openms.de
PEAKS	www.bioinform.com/download-peaks-studio
PepNovo+	proteomics.ucsd.edu/Software/PepNovo

PeptideAtlas	www.peptideatlas.org
PeptideProphet	peptideprophet.sourceforge.net
Percolator	percolator.ms
PRIDE	www.ebi.ac.uk/pride/archive
ProSight PTM	prosigthptm.northwestern.edu
ProteinProphet	proteinprophet.sourceforge.net
ProteomeXchange	www.proteomexchange.org
ProteoWizard	proteowizard.sourceforge.net
Proteomics Standards Initiative (PSI)	www.psidev.info
SEQUEST	www.proteomicswiki.com/wiki/index.php/SEQUEST_installation_instructions
SIMS	emililab.med.utoronto.ca
Tide	noble.gs.washington.edu/proj/tide
TopPIC	proteomics.informatics.iupui.edu/software/toppic
TPP	tools.proteomecenter.org/software.php
UniProt	www.uniprot.org
X! Hunter	ftp://ftp.thegpm.org/repos/xhunter
X! Hunter ASL	ftp://ftp.thegpm.org/proteotypic_peptide_profiles
X! Tandem	ftp://ftp.thegpm.org/projects/tandem

Further Reading

Nature Milestones in Mass Spectrometry (www.nature.com/milestones/milemassspec) is a collaborative effort involving five Nature Publishing Group journals. Each milestone article, written by a Nature Publishing Group editor, focuses on a key technical development in mass spectrometry covering one breakthrough. Each article highlights the main papers that contributed to the advance and the applications that stemmed from these advances.

References

- Aebersold, R. and Mann, M. (2003). Mass spectrometry-based proteomics. *Nature* 422 (6928): 198–207.
- Bauer, C., Cramer, R., and Schuchhardt, J. (2011). Evaluation of peak-picking algorithms for protein mass spectrometry. *Methods Mol. Biol.* 696: 341–352.
- Butterfield, D.A., Boyd-Kimball, D., and Castegna, A. (2003). Proteomics in Alzheimer's disease: insights into potential mechanisms of neurodegeneration. *J. Neurochem.* 86 (6): 1313–1327.
- Cox, J., Neuhauser, N., Michalski, A. et al. (2011). Andromeda: a peptide search engine integrated into the MaxQuant environment. *J. Proteome Res.* 10 (4): 1794–1805.
- Craig, R. and Beavis, R.C. (2004). TANDDEM: matching proteins with tandem mass spectra. *Bioinformatics* 20 (9): 1466–1467.
- Craig, R., Cortens, J.P., and Beavis, R.C. (2004). Open source system for analyzing, validating, and storing protein identification data. *J. Proteome Res.* 3 (6): 1234–1242.
- Deutsch, E.W., Csordas, A., Sun, Z. et al. (2017). The ProteomeXchange consortium in 2017: supporting the cultural change in proteomics public data deposition. *Nucleic Acids Res.* 45 (D1): D1100–D1106.
- Djuric, S.W., Hutchins, C.W., and Talaty, N.N. (2016). Current status and future prospects for enabling chemistry technology in the drug discovery process. *F1000Research* 5: 2426.

- Eng, J., McCormack, A., and Yates, J. (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: 976–989.
- Farrah, T., Deutsch, E.W., Hoopmann, M.R. et al. (2013). The state of the human proteome in 2012 as viewed through PeptideAtlas. *J. Proteome Res.* 12 (1): 162–171.
- Fenn, J.B., Mann, M., Meng, C.K. et al. (1989). Electrospray ionization for mass spectrometry of large biomolecules. *Science* 246 (4926): 64–71.
- Fenyö, D. (1999). The biopolymer markup language. *Bioinformatics* 15 (4): 339–340.
- Fenyö, D., Eriksson, J., and Beavis, R. (2010). Mass spectrometric protein identification using the global proteome machine. In: *Computational Biology* (ed. D. Fenyö), 189–202. Totowa, NJ: Humana Press.
- Filiou Michaela, D., Martins-de-Souza, D., Guest Paul, C. et al. (2012). To label or not to label: applications of quantitative proteomics in neuroscience research. *Proteomics* 12 (4–5): 736–747.
- Gaudet, P., Michel, P.A., Zahn-Zabal, M. et al. (2017). The neXtProt knowledgebase on human proteins: 2017 update. *Nucleic Acids Res.* 45 (D1): D177–D182.
- Gavin, A.C., Bosche, M., Krause, R. et al. (2002). Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature* 415 (6868): 141–147.
- Gerber, S.A., Rush, J., Stemman, O. et al. (2003). Absolute quantification of proteins and phosphoproteins from cell lysates by tandem MS. *Proc. Natl. Acad. Sci. U.S.A.* 100 (12): 6940–6945.
- Gramolini, A.O., Kislinger, T., Alikhani-Koopaei, R. et al. (2008). Comparative proteomics profiling of a phospholamban mutant mouse model of dilated cardiomyopathy reveals progressive intracellular stress responses. *Mol. Cell. Proteomics* 7 (3): 519–533.
- Gygi, S.P., Rist, B., Gerber, S.A. et al. (1999). Quantitative analysis of complex protein mixtures using isotope-coded affinity tags. *Nat. Biotechnol.* 17 (10): 994–999.
- Halligan, B.D., Geiger, J.F., Vallejos, A.K. et al. (2009). Low cost, scalable proteomics data analysis using Amazon's cloud computing services and open source search algorithms. *J. Proteome Res.* 8 (6): 3148–3153.
- Henzel, W.J., Billeci, T.M., Stults, J.T. et al. (1993). Identifying proteins from two-dimensional gels by molecular mass searching of peptide fragments in protein sequence databases. *Proc. Natl. Acad. Sci. U.S.A.* 90 (11): 5011–5015.
- Hsu, J.-L., Huang, S.-Y., Chow, N.-H., and Chen, S.-H. (2003). Stable-isotope dimethyl labeling for quantitative proteomics. *Anal. Chem.* 75 (24): 6843–6852.
- Huang, K.-Y., Lee, T.-Y., Kao, H.-J. et al. (2019). dbPTM in 2019: exploring disease association and cross-talk of post-translational modifications. *Nucleic Acids Res.* 47 (D1): D298–D308.
- Jennings, K.R. (1968). Collision-induced decompositions of aromatic molecular ions. *Int. J. Mass Spectrom. Ion Phys.* 1 (3): 227–235.
- Karas, M. and Hillenkamp, F. (1988). Laser desorption ionization of proteins with molecular masses exceeding 10,000 daltons. *Anal. Chem.* 60 (20): 2299–2301.
- Kelleher, N.L., Lin, H.Y., Valaskovic, G.A. et al. (1999). Top down versus bottom up protein characterization by tandem high-resolution mass spectrometry. *J. Am. Chem. Soc.* 121 (4): 806–812.
- Kislinger, T., Rahman, K., Radulovic, D. et al. (2003). PRISM, a generic large scale proteomic investigation strategy for mammals. *Mol. Cell. Proteomics* 2 (2): 96–106.
- Krogan, N.J., Cagney, G., Yu, H. et al. (2006). Global landscape of protein complexes in the yeast *Saccharomyces cerevisiae*. *Nature* 440 (7084): 637–643.
- Kwon, T., Choi, H., Vogel, C. et al. (2011). MSBlender: a probabilistic approach for integrating peptide identifications from multiple database search engines. *J. Proteome Res.* 10 (7): 2949–2958.
- Little, D.P., Speir, J.P., Senko, M.W. et al. (1994). Infrared multiphoton dissociation of large multiply charged ions for biomolecule sequencing. *Anal. Chem.* 66 (18): 2809–2815.

- Miyagi, M. and Rao, K.C.S. (2007). Proteolytic ^{18}O -labeling strategies for quantitative proteomics. *Mass Spectrom. Rev.* 26 (1): 121–136.
- Mukherjee, S., Stamatis, D., Bertsch, J. et al. (2017). Genomes OnLine Database (GOLD) v.6: data updates and feature enhancements. *Nucleic Acids Res.* 45 (D1): D446–D456.
- Muth, T., Peters, J., Blackburn, J. et al. (2013). ProteoCloud: a full-featured open source proteomics cloud computing pipeline. *J. Proteomics* 88 (Suppl C): 104–108.
- Olsen, J.V., Macek, B., Lange, O. et al. (2007). Higher-energy C-trap dissociation for peptide modification analysis. *Nat. Methods* 4 (9): 709–712.
- Ong, S.-E., Blagoev, B., Kratchmarova, I. et al. (2002). Stable isotope labeling by amino acids in cell culture, SILAC, as a simple and accurate approach to expression proteomics. *Mol. Cell. Proteomics* 1 (5): 376–386.
- Orchard, S., Hermjakob, H., and Apweiler, R. (2003). The proteomics standards initiative. *Proteomics* 3 (7): 1374–1376.
- Pandey, A. and Mann, M. (2000). Proteomics to study genes and genomes. *Nature* 405 (6788): 837–846.
- Perkins, D.N., Pappin, D.J.C., Creasy, D.M., and Cottrell, J.S. (1999). Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis* 20 (18): 3551–3567.
- Roepstorff, P. and Fohlman, J. (1984). Letter to the editors. *Biol. Mass Spectrom.* 11 (11): 601–601.
- Ross, P.L., Huang, Y.N., Marchese, J.N. et al. (2004). Multiplexed protein quantitation in *Saccharomyces cerevisiae* using amine-reactive isobaric tagging reagents. *Mol. Cell. Proteomics* 3 (12): 1154–1169.
- Schubert, O.T., Rost, H.L., Collins, B.C. et al. (2017). Quantitative proteomics: challenges and opportunities in basic and applied research. *Nat. Protoc.* 12 (7): 1289–1294.
- Schuit, F., Flamez, D., De Vos, A., and Pipeleers, D. (2002). Glucose-regulated gene expression maintaining the glucose-responsive state of beta-cells. *Diabetes* 51 (Suppl 3): S326–S332.
- Sinz, A. (2014). The advancement of chemical cross-linking and mass spectrometry for structural proteomics: from single proteins to protein interaction networks. *Expert Rev. Proteomics* 11 (6): 733–743.
- Stein, S. (1990). National Institute of Standards and Technology (NIST) Mass Spectral Database and Software, v.3.02. chemdata.nist.gov
- Syka, J.E.P., Coon, J.J., Schroeder, M.J. et al. (2004). Peptide and protein sequence analysis by electron transfer dissociation mass spectrometry. *Proc. Natl. Acad. Sci. U.S.A.* 101 (26): 9528–9533.
- Thompson, A., Schäfer, J., Kuhn, K. et al. (2003). Tandem mass tags: a novel quantification strategy for comparative analysis of complex protein mixtures by MS/MS. *Anal. Chem.* 75 (8): 1895–1904.
- Vizcaino, J.A., Csordas, A., del-Toro, N. et al. (2016). 2016 update of the PRIDE database and its related tools. *Nucleic Acids Res.* 44 (D1): D447–D456.
- Wilkins, M.R., Sanchez, J.-C., Gooley, A.A. et al. (1996). Progress with proteome projects: why all proteins expressed by a genome should be identified and how to do it. *Biotechnol. Genet. Eng. Rev.* 13 (1): 19–50.
- Zubarev, R.A., Kelleher, N.L., and McLafferty, F.W. (1998). Electron capture dissociation of multiply charged protein cations. A nonergodic process. *J. Am. Chem. Soc.* 120 (13): 3265–3266.

12

Protein Structure Prediction and Analysis

David S. Wishart

Introduction to Protein Structures

Over the last several chapters, we have seen how proteins can be conveniently represented and analyzed as character strings (sequences). Indeed, much of what we call bioinformatics today is based on using computers to manipulate, store, and compare sequences or character strings. However, it is important to remember that the field of bioinformatics encompasses more than just sequence analysis and that many of the most interesting and exciting applications in bioinformatics today are actually concerned with *structure* analysis – or, as it is sometimes called, *structural bioinformatics*. In fact, the origins of bioinformatics actually lie in the field of structural biology, as many of the first bioinformatic programs and the very first bioinformatic databases were developed to store, compare, and analyze protein structures (Bernstein et al. 1977; Hagen 2000). Interestingly, many of the concepts used in sequence analysis (such as archiving, aligning, and visualizing) actually have close parallels in structure analysis. However, the analysis of protein structures also has an added layer of challenges owing to their inherent complexity.

Proteins are perhaps the most complex chemical entities in nature. No other class of molecule (large or small) exhibits the variety of shapes, sizes, textures, and mobility that can be found in proteins. Proteins are so inherently complex that scientists have gone to great efforts to develop efficient methods to determine their structures, to visualize their shapes, to measure their motions, to simplify their descriptions, to compare their folds, and to look for underlying structural commonalities. In fact, the challenge of characterizing protein structures has been deemed so significant that, since 1960, more than a dozen Nobel prizes have been awarded to scientists who have determined or developed methods to characterize protein structures.

This chapter is intended to provide an overview of the bioinformatic tools and databases needed to analyze, archive, visualize, predict, and evaluate protein structures. It is organized into eight sections, providing a short introduction to protein structure, a brief review of how protein structures are determined, a summary of how protein structures are described, a description of the main protein structure databases, an overview of selected structure visualization tools, a description of bioinformatic tools for structure prediction, a summary of how proteins may be evaluated and, finally, a description of how proteins can be classified and compared.

How Protein Structures are Determined

Figure 12.1 provides a flow diagram describing how protein structures can be determined or “solved.” As can be seen from this diagram, there are three experimental techniques that can be used to generate detailed structural information about proteins at atomic resolution:

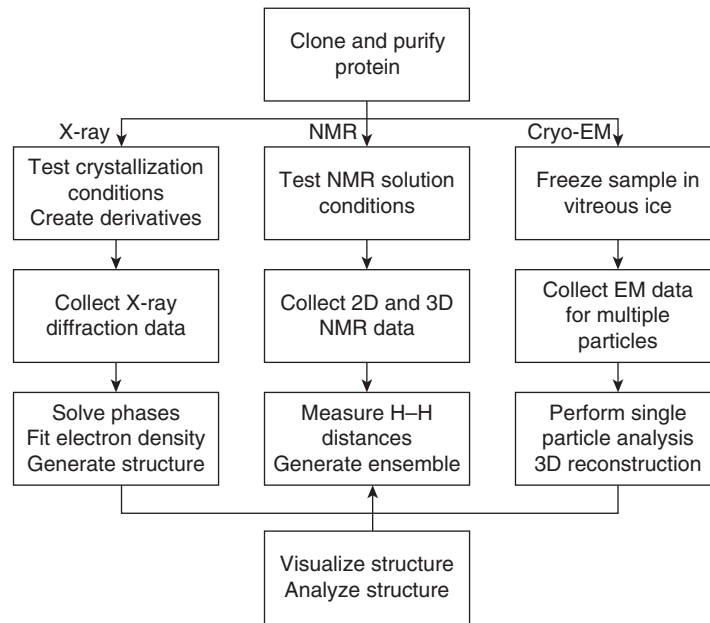


Figure 12.1 A flow diagram illustrating the steps used to experimentally prepare and solve (i.e. determine) the three-dimensional (3D) structures of proteins using X-ray, nuclear magnetic resonance (NMR), and cryogenic electron microscopy (cryo-EM) experimental techniques.

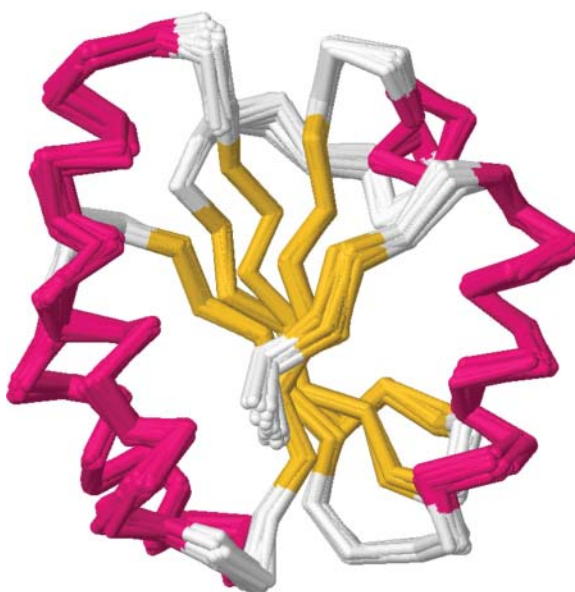
X-ray crystallography, nuclear magnetic resonance (NMR) spectroscopy, and electron microscopy. All protein structures must be determined from highly purified proteins that may be subsequently crystallized (for X-ray crystallography), placed in special solvents (for NMR spectroscopy), or frozen (for electron microscopy). X-ray crystallography is the oldest (and most precise) method, NMR is the next oldest (and least precise), and electron microscopy is the newest. In X-ray crystallography, small protein crystals (measuring less than 1 mm) are exposed to an intense beam of X-rays. The X-rays, which have a wavelength about the size of an atom (1–2 Å or ångstroms, which is 1×10^{-10} m), are scattered or diffracted by the protein atoms in the crystal. The diffraction pattern arising from this typically appears as tens of thousands of tiny spots arrayed in complex circular patterns. These diffraction patterns are recorded on a digital X-ray camera. The positions of the diffraction spots, along with their intensity (and some phase information), is actually sufficient for a computer to calculate an electron density map of all the heavy atoms – carbon, nitrogen, oxygen, sulfur – in the diffracting protein. From this map, crystallographers determine the x,y,z coordinates of all the atoms using the known sequence of the protein. Note that, in X-ray crystallography, even though the diffraction pattern arises from trillions of proteins contained in the crystal, the result is a structure for just a single “average” protein.

Protein crystallography is an experimentally challenging and computationally difficult process, so the brief synopsis given here does not do it justice. Excellent overviews of protein crystallography can be found in several high-quality textbooks (McCree 1999; Drenth 2006). The first X-ray structure of a protein (myoglobin) was determined in the late 1950s (Kendrew et al. 1958) and, since that time, more than 120 000 protein structures have been determined by X-ray techniques. X-ray crystallography permits the determination of very large macromolecular structures (hundreds of kilodaltons – even ribosomes and viruses), including both cytoplasmic and membrane-bound proteins. Recent computational, robotic, and instrumental advances (including the use of powerful synchrotrons) have made X-ray crystallography even more powerful and have greatly accelerated the structure determination process. Whereas it often took 6–7 years to purify, crystallize, and solve a protein structure in the 1970s, it is now possible (albeit rarely) to do it in as little as 6–7 days. As a result, more than 90% of all protein structures have been determined by X-ray crystallography.

Nevertheless, X-ray crystallography is not infallible. Crystallography, as the name implies, requires that proteins be studied in an “artificial” solid-state (crystalline) environment that does not resemble the normal physiological (aqueous) environment of the cell or body. As a result, the structures generated by X-ray crystallography are often altered by crystal packing and solvent exclusion effects. Likewise, not all parts of a protein (especially highly mobile regions) can be seen in an X-ray structure. Consequently, these “fuzzy” regions can be open to interpretation – or misinterpretation. It is also important to remember that X-ray structures of proteins are typically underdetermined, especially compared with X-ray structures of small molecules. The *R* factor (a measure of the agreement between the calculated structure and the experimental data) for “good” protein structures is typically 0.25, whereas, for small molecules, it is 0.05. Given that the highest *R* factor possible is 0.59 (for a completely wrong structure), one is led to the conclusion that even good protein structures are not without their faults. Indeed, it is not unusual for many protein structures to have some errors, ambiguities, or inaccuracies in their atomic positions ($\pm 0.5 \text{ \AA}$). Likewise, it is not unusual for a protein structure to be missing a few atoms or residues.

Compared with X-ray crystallography, NMR spectroscopy is a much newer (the first protein was “solved” in 1983) and somewhat more complicated technique. Therefore, a detailed explanation of the technique is beyond the scope of this chapter. An excellent overview of protein NMR can be found in a textbook written by Cavanagh et al. (2006). NMR is unique in that it allows one to study the structure and dynamics of molecules in a liquid state or in a near-physiological environment. In NMR spectroscopy, one determines protein structures not by measuring how X-rays are diffracted by atoms but, instead, by measuring how radio waves are absorbed by atomic nuclei such as hydrogen (^1H), isotopically labeled carbon (^{13}C), or nitrogen (^{15}N). This absorption measurement allows one to determine how much nuclear magnetism is transferred from one atom (or nucleus) to another. In NMR, this magnetization transfer is measured through chemical shifts, J-couplings, and nuclear Overhauser effects (NOEs). These parameters, which can be best observed for individual hydrogen atoms, must be determined for as many protein atoms as possible using complex multi-dimensional NMR experiments with whimsical acronyms such as COSY, TOCSY, NOESY, and HMQC. Once measured, these parameters define a set of approximate structural constraints that can be fed into a computer-based constraint minimization calculation (distance geometry or simulated annealing). The result is a series (15–50) of similar protein structures that satisfy the experimental constraints. Therefore, unlike X-ray methods that yield just one structure, NMR methods generate multiple structures – all of which are overlaid or superimposed on each other to produce so-called “blurrograms” (Figure 12.2). The quality of an NMR structure determination effort

Figure 12.2 An example of a nuclear magnetic resonance (NMR) “blurrogram” of a structure ensemble for *Escherichia coli* thioredoxin (Protein Data Bank database identifier: 4TRX). This represents a superposition of 33 near-identical structures of *E. coli* thioredoxin that satisfy all (or nearly all) of the measured NMR constraints.



is typically given by how closely matched these superimposed structures may be, with root mean square deviation (RMSD) values of $<1 \text{ \AA}$ being indicative of a good structure and RMSD values of $>2 \text{ \AA}$ being typical of a poorly determined structure (Box 12.1). Interestingly, these blurogram structures are probably more reflective of the true solution behavior of proteins as most proteins seem to exist in an ensemble of slightly different configurations.

Box 12.1 The Meaning of RMSD

Protein sequence alignments are evaluated in terms of an expect (E) value, a bit score, or percent identity. In the case of structure comparisons or structure alignments, these are often scored using a measure called root mean square deviation, or RMSD – something that is, interestingly enough, an archaic term for standard deviation. In other words, RMSD is calculated the same way a standard deviation is calculated. Once two structures are superimposed, the sum of the square of the differences in distance (in ångstroms, or \AA) between $C\alpha$ atoms is calculated and divided by the number of atoms compared. The square root of this number is called the RMSD, and it is normally reported in ångstroms. When more than two structures are superimposed, as is the case with NMR structure ensembles, a hypothetical average structure for the ensemble is first calculated, then the sum of the distance differences is calculated relative to this average structure. RMSD values are frequently used by NMR spectroscopists, structure modelers, and X-ray crystallographers when comparing structure ensembles, looking at related structures or characterizing structure families. Table 12.1 provides a rough guideline in terms of what a given RMSD value corresponds to in terms of structure quality for an NMR structure. The second column in the table provides a similar qualitative guideline for what RMSD values mean in terms of structure similarity.

Table 12.1 Relationship between backbone root mean square deviation (RMSD, in ångstroms) and structure quality for nuclear magnetic resonance (NMR) structure ensembles (column 1) and for protein structure comparisons (column 2).

RMSD (\AA)	NMR comment	Structure comparison comment
>12	Random coil	Completely unrelated
7.0	Major problems	Dubious relationship
5.0	Not quite converging	May be structurally related
4.0	Poor fit	Good structural relationship
2.0	Converging	Closely related
1.5	Barely acceptable	Very closely related
0.8	Typical NMR structure	Differences are not obvious
0.4	Best case NMR structure	Essentially indistinguishable

As there is no experimental requirement for crystals, NMR sample preparation is inherently easier than X-ray sample preparation. Furthermore, because NMR is a liquid-based system, NMR structures more likely resemble those seen in the normal physiological (liquid) environment of the cell or body. However, NMR is often limited by the size of the molecule being studied (the practical upper limit is $\sim 40 \text{ kDa}$), the solubility of the molecule (membrane proteins cannot be studied), and the requirement for special isotopically labeled molecules (expensive). Furthermore, NMR structures are inherently less precise than X-ray structures. Continuing computational and instrumental improvements have made NMR much easier and much faster than ever before. Indeed, it is now possible to determine a protein NMR structure in a few weeks. Approximately 10% of all known protein structures have been determined by NMR.

The most recent addition to the structural biologist's arsenal of tools is cryogenic electron microscopy (cryo-EM or three-dimensional [3D] cryo-EM). Unlike NMR spectroscopy or X-ray

crystallography, both of which are “indirect” methods requiring sophisticated mathematics to transform complex X-ray diffraction or NMR absorption data into structural information, cryo-EM is a direct technique. In other words, what you see is what you get. Direct visualization of atomic structures has always been a dream for structural biologists and cryo-EM now offers that possibility. In cryo-EM, protein samples are quickly frozen in water (creating vitreous ice) and then placed under powerful electron beams with electron wavelengths of 1–2 Å. By using newly developed electronic optics called phase plates, better and more sensitive detection systems, very fast “freeze frame” data collection methods and sophisticated image averaging, it is now possible to determine protein structures with atomic-level resolution quite routinely (Bai et al. 2015). Cryo-EM sample preparation is much easier than sample preparation for X-ray crystallography and cryo-EM structures likely resemble those seen in the normal liquid environment of the cell. Like NMR spectroscopy, cryo-EM is limited by the size of the molecule being studied – except in the opposite way. Big proteins (>100 kDa) are preferred, as small molecules are typically too small to be seen (although this is changing). Other than the size restriction, cryo-EM has relatively few limitations. Indeed, some cryo-EM structures are now even more precisely determined than X-ray structures. While only 1% of known protein structures have been solved by cryo-EM, rapid computational and instrumental improvements are making protein structure determination by cryo-EM the preferred route for many structural biologists. Indeed, the 2017 Nobel Prize in Chemistry was awarded to Jacques Dubochet, Joachim Frank, and Richard Henderson for “developing cryo-electron microscopy for the high-resolution structure determination of biomolecules in solution.”

How Protein Structures are Described

Today, the most common approach to describing protein structures is known as the hierarchical method. In this schema, a protein is viewed as having different “levels” of structure with progressively greater complexity (Figure 12.3). The simplest level is called the primary structure. By definition, a protein’s primary structure is simply its amino acid sequence. Of course, proteins are not just letters printed on a page. In reality, they are made of different combinations of amino acids covalently connected together by peptide bonds. The resulting polymer

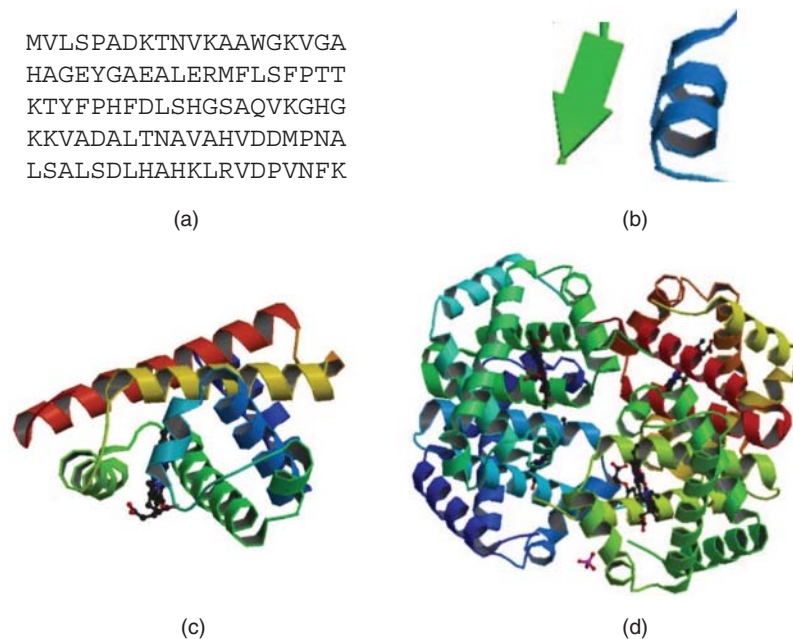


Figure 12.3 The different levels of protein structures illustrating: (a) primary structure; (b) secondary structure; (c) tertiary structure; and (d) quaternary structure for hemoglobin.

exhibits much of the chain-like flexibility and behavior as most other polymers. However, the partial double bond character of each peptide bond, the varying chemical nature of the different amino acid side chains, along with the steric restrictions imposed by the presence of these side chains means that proteins do not (or cannot) exist as an extended string of amino acids. In other words, proteins have a natural proclivity to fold up and form more complex structures.

The next level up in the structural hierarchy is called secondary structure (Figure 12.3b). Secondary structures are defined as the repetitive hydrogen-bonded shapes or substructures that make up sequentially proximal components of proteins. Some of the most common protein secondary structures are helices (~35% of all residues) and beta-pleated sheets (~25% of all residues). Both kinds of secondary structures were originally predicted by Linus Pauling in the 1950s (Corey and Pauling 1953). These structures are characterized by regular hydrogen bonding patterns that persist over three or more consecutive residues. In addition to these two very abundant forms of secondary structure, there also several other kinds of less abundant but still important secondary structures, including beta turns (sharp chain reversals), omega loops (characterized by loops having a shape resembling the Greek letter omega [Ω]), and 3/10 helices. Together, these five general classes of secondary structure can be routinely assigned (manually or automatically) to about 55–65% of all the amino acids in proteins (Willard et al. 2003). The remaining unclassified or unclassifiable substructures are typically called random coil or, more properly, unstructured regions.

By assembling different pieces of secondary structure together it is possible to create a complete protein structure. This assemblage of different secondary structural components is called the tertiary structure (Figure 12.3c). Tertiary structure is just another term for the 3D structure of a protein. Unlike secondary structure, tertiary structure is primarily determined or mediated by hydrophobic interactions between distal parts of the polypeptide chain. Just as with secondary structures, there are several different classes or groupings of tertiary structures. These classes have been identified by careful inspection of thousands of X-ray and NMR structures by skilled structural biologists and bioinformaticians. The simplest tertiary structure classification scheme refers to the relative content of different secondary structure elements (Levitt and Chothia 1976). This includes the all-alpha (>50% helix; <10% beta sheet), all-beta (>30% beta sheet; <5% helix), and mixed or alpha/beta (everything else) structural classes. More refined tertiary classification schemes exist that take into account common topologies, motifs, or folds found in a large number of non-homologous proteins. Common tertiary folds include the α/β barrel (superoxide dismutase); the four-helix bundle (cytochrome C550); the Greek key (immunoglobulins); the E-F hand (calcium binding proteins); the zinc finger, and so on. Some examples of these protein folds are shown in Figure 12.4. Among the 120 000 protein structures that have been solved so far, approximately 1200–1300 distinct “folds” have been identified. What is particularly intriguing (and exciting!) is that this number is very close to the predicted number of all biologically feasible protein folds, which is about 1500 (Levitt

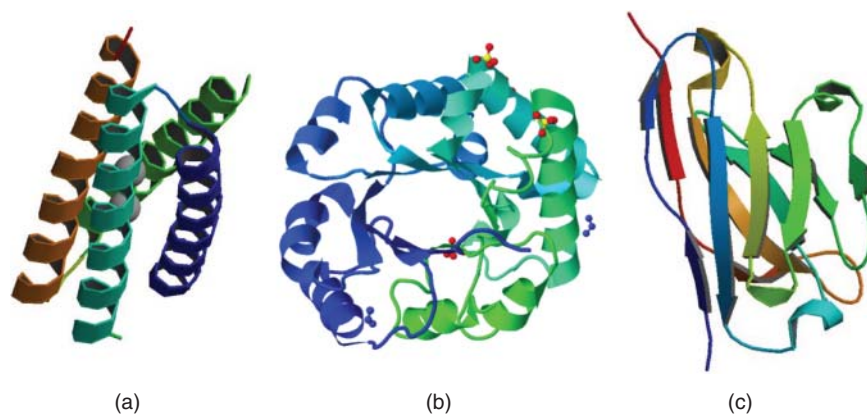


Figure 12.4 Examples of different types of protein folds including (a) the four helix bundle; (b) the alpha-beta barrel; (c) the immunoglobulin fold.

2007; Schaeffer and Daggett 2011). This suggests we may not be too far from creating a kind of “periodic table” of protein structures or substructures.

Beyond the tertiary structure level is something called quaternary structure (Figure 12.3d). Quaternary structure refers to the assemblage of two or more independent tertiary structures into a larger superstructure such as the two chains of insulin, the four chains of hemoglobin, or the 50+ peptide chains in bacterial ribosomes. Many proteins need to form quaternary complexes to function, and so understanding or identifying quaternary structure is key to understanding protein–protein interactions (see Chapter 13).

Protein structures are almost always described in terms of Cartesian (i.e. x,y,z) coordinates of the constituent atoms using a standard format known as the Protein Data Bank (PDB) format (Box 12.2). X-ray and cryo-EM structure files are limited to containing coordinates of only the heavy atoms (C, N, O, and S), while NMR structure files typically contain both the heavy atoms and the attached hydrogen atoms as well. Most protein data files will have several thousand atoms and, therefore, several thousand lines and several thousand coordinate positions associated with each atom. As all proteins are made up of amino acids, there is a relatively standard geometry for each atom in each amino acid – that is, every atom is at a well-defined bonding distance or bond angle relative to every other atom (Figure 12.5). As seen in this figure, each amino acid is made up of a nitrogen (N) atom bonded to a central carbon atom ($C\alpha$), with the N– $C\alpha$ bond being 1.47 Å in length. Likewise the distance between the $C\alpha$ atom and the carbonyl atom (C) is 1.53 Å, while the distance between the carbonyl carbon (C) and its oxygen (O) is 1.24 Å. The central $C\alpha$ atom is also connected to a central hydrogen atom ($H\alpha$) that is 1.00 Å away and to a side chain carbon (R or $C\beta$) that is 1.56 Å away. The peptide N–C bond is always 1.32 Å in length. Given this geometric consistency, it is actually possible to describe protein structures in terms of internal coordinates or internal angles instead of Cartesian coordinates. Internal coordinates are coordinates that do not need or are not defined by an origin. By using a class of planar angles called “dihedral” angles (Figure 12.5) – also known as torsion angles – it is possible to compactly describe the backbone or general topology of protein structures. The two most important backbone dihedral angles are the angle defined by an amino acid residue’s H, N, $C\alpha$, and $H\alpha$ atoms (called phi or ϕ) and the angle defined by an amino acid residue’s $H\alpha$, $C\alpha$, C, and O atoms (called psi or ψ). In other words, the ϕ angle is along the N– $C\alpha$ bond, while the ψ angle is along the $C\alpha$ –C bond. Each residue in a protein can be defined by one ϕ and one ψ angle. Therefore, the entire protein backbone can be defined by the set of all ϕ/ψ angles for all residues in the protein.

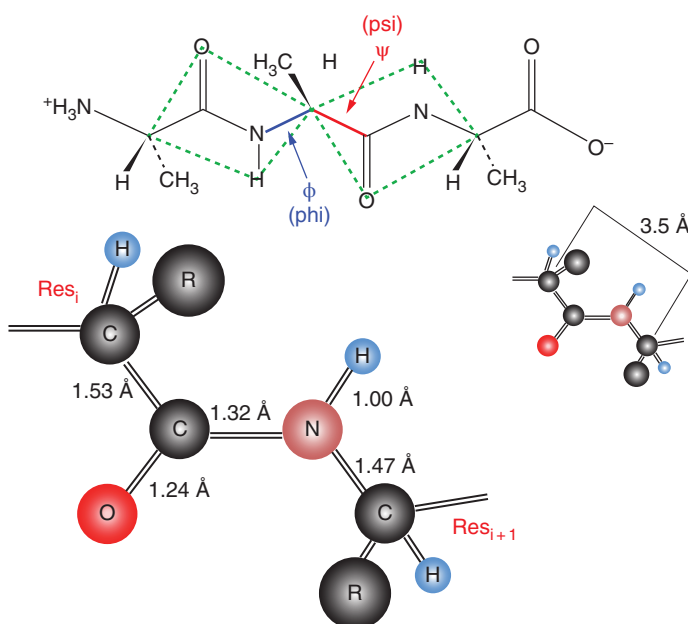


Figure 12.5 An illustration of standard amino acid residue and peptide bond geometry. Typical bond lengths are shown along with standard backbone dihedral angles.

Box 12.2 PDB Format

The standard format for protein structure files is called the Protein Data Bank (PDB) format (Figure 12.6). This is a machine- and human-readable format that allows information about the protein, the depositors, the sequence, the secondary structure, and the x,y,z coordinates to be stored and read by a computer. The PDB format is quite old, reflecting the state of computation in the late 1970s when the PDB was established. As a result, each line in the PDB file must have a seven-letter (or less) tag, followed by an exact number of spaces, which in turn is followed by the information (all in upper case) relevant to that tag. Each PDB file is structured almost identically with the first few lines (labeled by HEADER, CMPND, SOURCE, AUTHOR, or JRNL), where HEADER provides the protein function, PDB ID, and deposition date, CMPND provides the protein name, SOURCE provides the source organism, AUTHOR lists the authors, and JRNL lists the journal where the structure was published, respectively. The next set of lines (labeled REMARK) provide additional details on the resolution, R factor (a quality measure), methods used to solve the structure, number of molecules in the asymmetric unit, and so on, mostly in free format. The sequence is presented (labeled SEQRES) using the now archaic three-letter amino code, followed by HET and FORMUL labels indicating the names and chemical formulae of hetero-atoms (non-amino acid moieties found in the structure). The secondary structure, as identified by the depositor, is indicated by HELIX, SHEET, TURN, and SSBOND tags.

These first 100 or so lines constitute the “header” of a PDB file and provide a useful overview of the protein and the quality of the structure. The next set of lines in a PDB file provide the atomic coordinates. These are always labeled by the ATOM tag. Up to 10 columns of text and numbers follow each ATOM tag including the atom number, atom label (CA = alpha carbon, C = carbonyl carbon, etc.), residue name (three-letter code), chain number or letter, residue number, X coordinate (in ångströms), Y coordinate (in ångströms), Z coordinate (in ångströms), occupancy (usually 1.00), and thermal B factor (a measure of mobility).

The PDB format, while generally easy for a human to read, can be quite confounding for a computer to read. For instance, there are frequent exceptions and variations in the labeling, numbering, and formatting of many PDB files, particularly those deposited prior to 1995. Furthermore, the programs that read PDB formatted files must have certain chemical knowledge built into them – that is, the connections and bonds between atoms must be known (or inferred), as this connectivity information is not provided in the PDB file. Additionally, there is no formal data dictionary that describes all the rules for writing or reading a PDB file. This makes writing programs to handle, analyze, and view PDB files quite a challenge.

Given the inconsistencies, informality, and archaic nature of the PDB format, there have been a number of efforts to correct or migrate PDB files to a more consistent and modern file format. Over the past number of years, the PDB has been storing its files (internally) in a format called mmCIF (for macromolecular Crystallographic Information File) that is based on the CIF format used by small-molecule crystallographers (Hall et al. 1991). The mmCIF format is a simple and consistent data representation for exchanging and archiving structural data that is endorsed by a number of international agencies. As of 2011, the PDB now maintains all of its data in the PDBML/XML format. PDBML stands for PDB Markup Language (Westbrook et al. 2005). This newer format provides a representation of PDB data in XML format according to the PDBx/mmCIF Exchange Data Dictionary. However, because so many software packages have already been written to handle PDB formatted files and relatively few have been written to handle PDBML, it is likely that the legacy PDB format will be around for a long, long time to come.

```

HEADER      ELECTRON TRANSPORT                19-MAR-90   2TRX   2TRXA  1
COMPND      THIOREDOXIN                        2TRXA  2
SOURCE      (ESCHERICHIA $COLI)                 2TRX   4
AUTHOR      S.K.KATTI,D.M.LE*MASTER,H.EKLUND   2TRX   5
JRNL        AUTH  S.K.KATTI,D.M.LE*MASTER,H.EKLUND 2TRX   7
JRNL        TITL  CRYSTAL STRUCTURE OF THIOREDOXIN FROM ESCHERICHIA 2TRX   8
JRNL        TITL 2 $COLI AT 1.68 ANGSTROMS RESOLUTION 2TRX   9
JRNL        REF   J.MOL.BIOL.                   V. 212   167 1990 2TRX  10
JRNL        REFN  ASTM JMOBAK  UK ISSN 0022-2836      070 2TRX  11
REMARK      2                                     2TRX  31
REMARK      2 RESOLUTION. 1.68 ANGSTROMS.          2TRX  32
REMARK      3                                     2TRX  33
REMARK      3 REFINEMENT. BY THE RESTRAINED LEAST-SQUARES PROCEDURE OF J. 2TRX  34
REMARK      3 KONNERT AND W. HENDRICKSON AS MODIFIED BY B. FINZEL 2TRX  35
REMARK      3 (PROGRAM *PROFFT*). THE R VALUE IS 0.165 FOR 25969 2TRX  36
REMARK      3 REFLECTIONS IN THE RESOLUTION RANGE 8.0 TO 1.68 ANGSTROMS 2TRX  37
REMARK      3 WITH FOBS .GT. 3.0*SIGMA(FOBS) 2TRX  38
REMARK      3                                     2TRX  39
SEQRES      1 A 108 SER ASP LYS ILE ILE HIS LEU THR ASP ASP SER PHE ASP 2TRX  74
SEQRES      2 A 108 THR ASP VAL LEU LYS ALA ASP GLY ALA ILE LEU VAL ASP 2TRX  75
SEQRES      3 A 108 PHE TRP ALA GLU TRP CYS GLY PRO CYS LYS MET ILE ALA 2TRX  76
SEQRES      4 A 108 PRO ILE LEU ASP GLU ILE ALA ASP GLU TYR GLN GLY LYS 2TRX  77
SEQRES      5 A 108 LEU THR VAL ALA LYS LEU ASN ILE ASP GLN ASN PRO GLY 2TRX  78
SEQRES      6 A 108 THR ALA PRO LYS TYR GLY ILE ARG GLY ILE PRO THR LEU 2TRX  79
SEQRES      7 A 108 LEU LEU PHE LYS ASN GLY GLU VAL ALA ALA THR LYS VAL 2TRX  80
SEQRES      8 A 108 GLY ALA LEU SER LYS GLY GLN LEU LYS GLU PHE LEU ASP 2TRX  81
SEQRES      9 A 108 ALA ASN LEU ALA 2TRX  82
HET  MPD    606      8  2-METHYL-2,4-PENTANEDIOL 2TRX 107
HET  MPD    607      8  2-METHYL-2,4-PENTANEDIOL 2TRX 108
HET  MPD    608      8  2-METHYL-2,4-PENTANEDIOL 2TRX 109
FORMUL      3  CU    2(CU1 ++)) 2TRX 110
FORMUL      4  MPD    8(C6 H14 O2) 2TRX 111
FORMUL      5  HOH   *140(H2 O1) 2TRX 112
HELIX       1  A1A SER A  11  LEU A  17  1  DISORDERED IN MOLECULE B 2TRX 113
HELIX       2  A2A CYS A  32  TYR A  49  1  BENT BY 30 DEGREES AT RES 39 2TRX 114
HELIX       3  A3A ASN A  59  ASN A  63  1  2TRX 115
HELIX       4  31A THR A  66  TYR A  70  5  DISTORTED H-BONDING C-TERMINS 2TRX 116
HELIX       5  A4A SER A  95  LEU A 107  1  2TRX 117
HELIX       6  A1B SER B  11  LEU B  17  1  DISORDERED IN MOLECULE B 2TRX 118
SSBOND      1  CYS A  32    CYS A  35  2TRX 143
ATOM        1  N    SER A  1    21.389  25.406  -4.628  1.00 23.22 2TRX 152
ATOM        2  CA   SER A  1    21.628  26.691  -3.983  1.00 24.42 2TRX 153
ATOM        3  C    SER A  1    20.937  26.944  -2.679  1.00 24.21 2TRX 154
ATOM        4  O    SER A  1    21.072  28.079  -2.093  1.00 24.97 2TRX 155
ATOM        5  CB   SER A  1    21.117  27.770  -5.002  1.00 28.27 2TRX 156
ATOM        6  OG   SER A  1    22.276  27.925  -5.861  1.00 32.61 2TRX 157
ATOM        7  N    ASP A  2    20.173  26.028  -2.163  1.00 21.39 2TRX 158
ATOM        8  CA   ASP A  2    19.395  26.125  -0.949  1.00 21.57 2TRX 159
ATOM        9  C    ASP A  2    20.264  26.214  0.297  1.00 20.89 2TRX 160
ATOM       10  O    ASP A  2    19.760  26.575  1.371  1.00 21.49 2TRX 161
ATOM       11  CB   ASP A  2    18.439  24.914  -0.856  1.00 22.14 2TRX 162
ATOM       22  CE   LYS A  3    21.620  21.104  2.844  1.00 25.84 2TRX 173
ATOM       23  NZ   LYS A  3    20.830  20.757  1.615  1.00 25.55 2TRX 174

```

Figure 12.6 An example of a Protein Data Bank formatted file showing the first ~50 lines of the *Escherichia coli* thioredoxin entry (Protein Data Bank database identifier: 2TRX).

Interestingly, if these ϕ/ψ torsion angles are plotted (for known protein structures) with ϕ on the horizontal (X) axis and ψ on the vertical (Y) axis, a clear distribution can be seen (Figure 12.7). This kind of graph is called a Ramachandran plot (Ramachandran et al. 1963), and was developed by the Indian crystallographer Gopalasamudram Narayana Ramachandran. Empty regions in this Ramachandran plot (accounting for ~75% of the area)

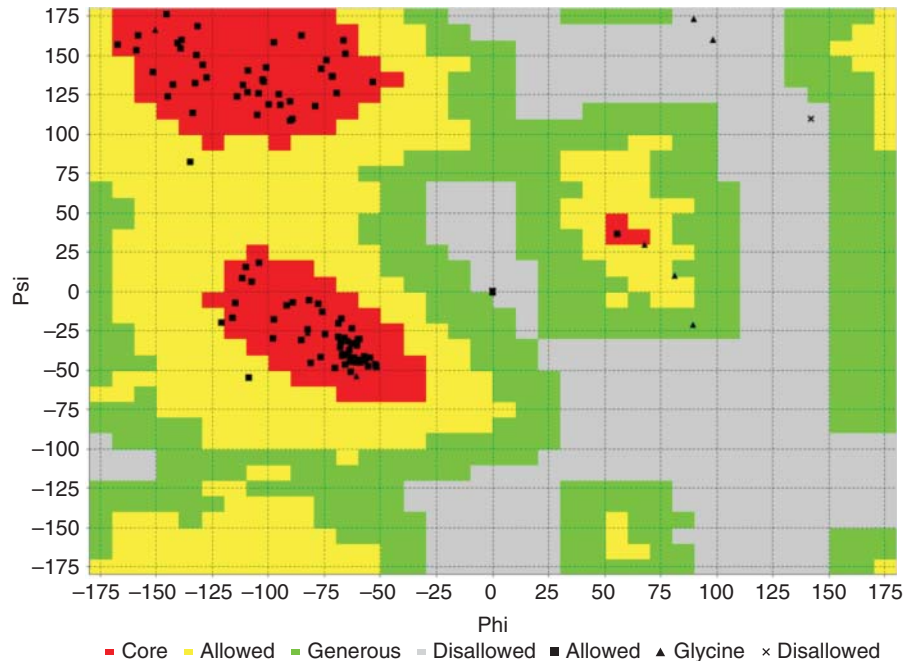


Figure 12.7 A Ramachandran plot for the thioredoxin protein (Protein Data Bank database identifier: 2TRX) as generated using the program VADAR (Willard et al. 2003). Each black point in the plot corresponds to a single residue in the protein. A square corresponds to a residue in the “allowed” or “core” region, a triangle corresponds to a glycine residue, and an “X” corresponds to a residue in a “disallowed” region (see key). The “core boundaries” or red regions on the plot delineate the areas on the Ramachandran plot where ~85% of residues should be found in good quality structures. The “allowed boundaries” (green regions) delineate the portion of the plot where ~10% of residues should be found. Residues falling in the “generously allowed boundaries” (yellow regions) or outside this region indicate residues that may have serious steric problems. Glycine residues (marked with an “X”) are the exception as they can appear anywhere in the plot.

indicate where steric clashes of the amino acid side chains prevent these torsion angles from being accessed. When amino acid residues have torsion angles in the upper left quadrant (centered around $\phi = -120^\circ$ and $\psi = 120^\circ$) of the Ramachandran plot, they are in beta strands. When amino acids are in the lower left quadrant (centered around $\phi = -60^\circ$ and $\psi = -40^\circ$), they are in alpha helices. Ramachandran plots have found considerable utility in assessing the quality of protein structures. By studying a large number of high-quality structures and looking at their Ramachandran plots, it has been discovered that very good structures exhibit very tight clustering patterns and that relatively few residues are found to lie outside these tight clusters or “allowed” dihedral regions (Laskowski et al. 1993). Protein structures that are found to have a high percentage (>15%) of non-glycine residues in disallowed regions inevitably are found to be poor quality structures. Given their utility and simplicity, many protein structure software packages now include Ramachandran plots as part of their structure visualization and evaluation tools (Laskowski et al. 1993; Willard et al. 2003).

While protein structures can be described in terms of torsion angles, most representations still use Cartesian coordinates. However, it is important to remember that proteins are not simply composed of point-like atoms with point-like x,y,z coordinates. Indeed, atoms and amino acids occupy space or volume. Consequently, proteins have volumes and shapes. These shapes also have surfaces (both exterior and interior), which are defined by a surface area. These surfaces are quite rough and convoluted, and it is this surface “roughness” that gives proteins their unique properties, including their ligand binding sites or their protein interaction sites. Not all protein surfaces are accessible by other molecules or other atoms, so protein surfaces are often defined by their so-called accessible surface area, or ASA (Richards 1977). As nitrogen and oxygen atoms also carry partial charges, the atomic surfaces of proteins can also have positive

or negative charges, which attract opposite charges or repel like charges. Uncharged atoms (such as C) are typically hydrophobic and hydrophobic atoms will tend to attract each other. This “volumetric,” space-filling view of proteins is quite important when thinking about how proteins work and how they fold. More details about how proteins can be rendered and viewed will be described in Visualizing Proteins.

Protein Structure Databases

Just as sequence databases serve as the foundation to most elements of conventional bioinformatics, structural databases are the foundation to all of structural bioinformatics. Indeed, the very first electronic database in bioinformatics was a protein structure database – the PDB. The PDB was originally set up at the Brookhaven National Laboratory by Walter Hamilton in 1971 (Westbrook et al. 2003). When this effort began, there were just seven protein structures in the PDB and their coordinates were stored and distributed on punch cards and computer tape. Since then, the PDB has grown to contain more than 120 000 experimentally determined 3D structures of proteins, nucleic acids, carbohydrates, and their complexes. Furthermore, coordinate distribution and deposition is no longer done on punch cards – now it is done over the web. To cope with these changes and the rapid expansion of the database, the management of the PDB was moved from Brookhaven to the Research Collaboratory for Structural Bioinformatics (RCSB) in 1998, which included Rutgers University, the San Diego Supercomputing Center, and the Scripps Research Institute.

In 2003, the RCSB-PDB joined with other protein structure databases in Europe and Asia, including the European Macromolecular Structure Database or MSD-EBI (now known as PDBe) and the Japanese version of the PDB (known as PDBj), to create the Worldwide PDB, or wwPDB. In 2006, the wwPDB was expanded to include the BioMagResBank (BMRB), incorporating NMR structure and NMR experimental data. Currently, the PDB archive is managed by the Worldwide PDB Organization, which now includes the US regional data center (RCSB-PDB), the European regional data center (PDBe), the Japanese regional data center (PDBj), and the global NMR data center (BMRB). Each of these data centers annotates, validates, and disseminates standardized PDB data to the rest of the world. Each center also exchanges data with the other center on a daily basis so that all PDB files are accessible to users no matter which center they choose to work with. While the coordinate data files are always identical, each of the regional data centers is free to develop its own interfaces or to incorporate its own unique analysis/visualization tools. Some centers (such as PDBe and RCSB-PDB) offer exceptionally useful tools that are unique to their web sites. To simplify the discussion here, we will just focus on describing the tools and resources available through the RCSB-PDB.

Just like GenBank and the European Molecular Biology Laboratory (EMBL), the RCSB-PDB web site supports a number of services for submitting, searching, retrieving, and viewing data. The PDB equivalent of GenBank’s BankIt or Sequin (for DNA sequence submission) is known as OneDep (Young et al. 2017). OneDep is a uniform web deposition and annotation system that works for the RCSB-PDB, PDBe, PDBj, and BMRB. It allows structural biologists to directly deposit structural coordinates, electron density maps, or experimental NMR data and to have both the format checked (to match the PDBx/mmCIF master file format; Box 12.2) and the structural data automatically validated using specific validation tools for X-ray, cryo-EM, or NMR data. For X-ray structures, bond lengths, bond angles, bond planes, backbone torsion angles, side chain conformations, all-atom contacts, packing, hydrogen bond quality, structure factor data, and Wilson plots (used to determine the absolute scale of the diffracted intensities and to find the temperature factor) are all assessed through the wwPDB validation suite (Read et al. 2011). This suite includes structure quality checking software, such as PROCHECK (Laskowski et al. 1993), WHAT_CHECK (Hooft et al. 1996), MolProbity (Davis et al. 2007), RosettaHoles2 (Sheffler and Baker 2010), and SFCHECK (for structure factors; Vaguine et al.

1999). For NMR-derived structures, similar kinds of structure checks are performed, but additional chemical shift and NOE assessments are also performed using tools such as PANAV (Wang et al. 2010), SHIFTX2 (Han et al. 2011), and CING (Doreleijers et al. 2012). Through OneDep, inconsistencies, format problems, or questionable data are reported to the submitters, who must then make the requested corrections.

Prior to the introduction of structure validation tools, there were a number of examples of seriously flawed and “fake” structures deposited into the PDB (Lüthy et al. 1992; Hooft et al. 1996; Borrell 2009). Unlike the situation for DNA sequencing, where a sequencing ambiguity can be identified by re-sequencing in a matter of hours or days, problems with structures can take months or years to resolve. Hence, automatic structural validation is a particularly important service for structural biologists and one in which bioinformaticians have played a crucial role.

Of course, most PDB users are not interested in depositing data, but rather in accessing them. The RCSB-PDB is constantly evolving its user interface, so a screenshot or a detailed explanation of the RCSB-PDB homepage will always be out of date. However, there are a few constants. The RCSB-PDB has always supported standard text searches that allow users to query the PDB for matching PDB identifier (ID) codes. The PDB ID code is analogous to the GenBank accession number. It is a four-character identifier in which the first character is always a number between 1 and 9. Users may also search the PDB using protein names, sequences, authors, or any Boolean (AND, OR) combination. An advanced search is also available that allows users to search the PDB on the basis of more than 100 data fields including keywords, structure annotations, structure or sequence features, chemical components, authors, deposition dates, release dates, sequences (via FASTA; see Chapter 3), secondary structure content, resolution, space group, and so on. It is also possible to customize the resulting display in terms of what is shown and how it is ordered. The ability to search the PDB on the basis of a FASTA-formatted sequence query is particularly useful.

Results from any PDB search are displayed in the PDB Structure Summary page(s) (Figure 12.8). These pages provide users with a high-level overview of the protein structure, including information about the deposition title, depositor names, deposition and release dates, source organism, and brief details concerning the method of structure determination. From the Structure Summary pages, users may also download PDB files, the primary citation, and the wwPDB validation report (an assessment of the structure quality). The PDB Structure Summary pages typically have tabs or hyperlinks to interactive web-based 3D structure viewers, protein domain and functional annotations, sequence, secondary structure, and binding site data; sequence and structural homologs, experimental structure determination details, and related references and citations. These tabs or hyperlinks will typically have shortened titles to facilitate quick navigation.

Through the Structure Summary pages, users can access several different macromolecular viewing packages. These choices are always changing to reflect the state of the field, with the most recent ones including the JavaScript viewers NGL Viewer (Rose and Hildebrand 2015), JSmol (Hanson et al. 2013), and PV (Protein Viewer). They also include several stand-alone Java-based or Java applet viewers such as Simple Viewer, Protein Workshop, Ligand Explorer, and Kiosk Viewer. Java applets have been the mainstay for structure visualization on the web for about two decades. However, with the advent of increased internet security precautions, Java applet viewers have become quite cumbersome, and many web browsers (such as Chrome) no longer support Java applets. As a result, the PDB, along with many other data resources, are actively promoting JavaScript viewers such as NGL Viewer and JSmol. An example of a protein image generated via JSmol is shown in Figure 12.9. JSmol is fast, easy to use, and very convenient. NGL Viewer is another lightweight, highly scalable JavaScript viewer that can render large molecular complexes (millions of atoms) on just about any platform, including smartphones. NGL Viewer is particularly fast because it uses the compact,

RCSB PDB – 2TRX: CRYSTAL STRUCTURE OF THIOREDOXIN FROM ESCHERICHIA COLI AT 1.68 ANGSTROMS RESOLUTION Structure Summary Page

www.rcsb.org/pdb/explore.do?structureId=2TRX

RCSB PDB Deposit Search Visualize Analyze Download Learn More MyPDB Login

RCSB PDB An Information Portal to 129942 Biological Macromolecular Structures

Search by PDB ID, author, macromolecule, sequence, or ligand Go

Advanced Search | Browse by Annotations | Search History (22) | Previous Results (86)

Structure Summary 3D View Annotations Sequence Sequence Similarity Structure Similarity Experiment

Literature

Biological Assembly 1

2TRX

CRYSTAL STRUCTURE OF THIOREDOXIN FROM ESCHERICHIA COLI AT 1.68 ANGSTROMS RESOLUTION

DOI: 10.2210/pdb2trx/pdb

Classification: ELECTRON TRANSPORT

Deposited: 1990-03-19 Released: 1991-10-15

Deposition author(s): Katti, S.K., Lemaster, D.M., Eklund, H.

Organism: Escherichia coli

Structural Biology Knowledgebase: 2TRX (>16 annotations) SBKB.org

Experimental Data Snapshot

Method: X-RAY DIFFRACTION

Resolution: 1.68 Å

R-Value Observed: 0.165

wwPDB Validation

Metric	Percentile Ranks	Value
Clashscore	5	5
Ramachandran outliers	0.9%	0.9%
Sidechain outliers	1.7%	1.7%

Literature

Download Primary Citation

Crystal structure of thioredoxin from Escherichia coli at 1.68 A resolution.

Katti, S.K., LeMaster, D.M., Eklund, H.

(1990) J. Mol. Biol. 212: 167-184

PubMed: 2181145 Search on PubMed

DOI: 10.1016/0022-2836(90)90313-B

PubMed Abstract:

The crystal structure of thioredoxin from Escherichia coli has been refined by the

Figure 12.8 A screenshot of the Research Collaboratory for Structural Bioinformatics (RCSB)–Protein Data Bank (PDB) Structure Summary page for *Escherichia coli* thioredoxin.

binary, Macromolecular Transmission Format (MMTF) to save time loading files over the internet. In addition to these interactive 3D viewers, the RCSB-PDB Structure Summary page also provides a number of good quality still images showing both the asymmetric unit (the structure as it appears in the crystal) and different biological assemblies (the active version of the molecule, including its quaternary structure). These still images use a rainbow color gradient to help identify the N-terminus (blue) from the C-terminus (red).

While interactive structure visualization is a particularly appealing feature of the RCSB-PDB, there are also a number of important additional services or offerings that are available. In particular, the RCSB-PDB provides pre-calculated lists and links to structural homologs through its Structure Similarity link (Prlic et al. 2010). As we will see later in this chapter, the identification of structural homologs is a much more computationally complicated and time-consuming

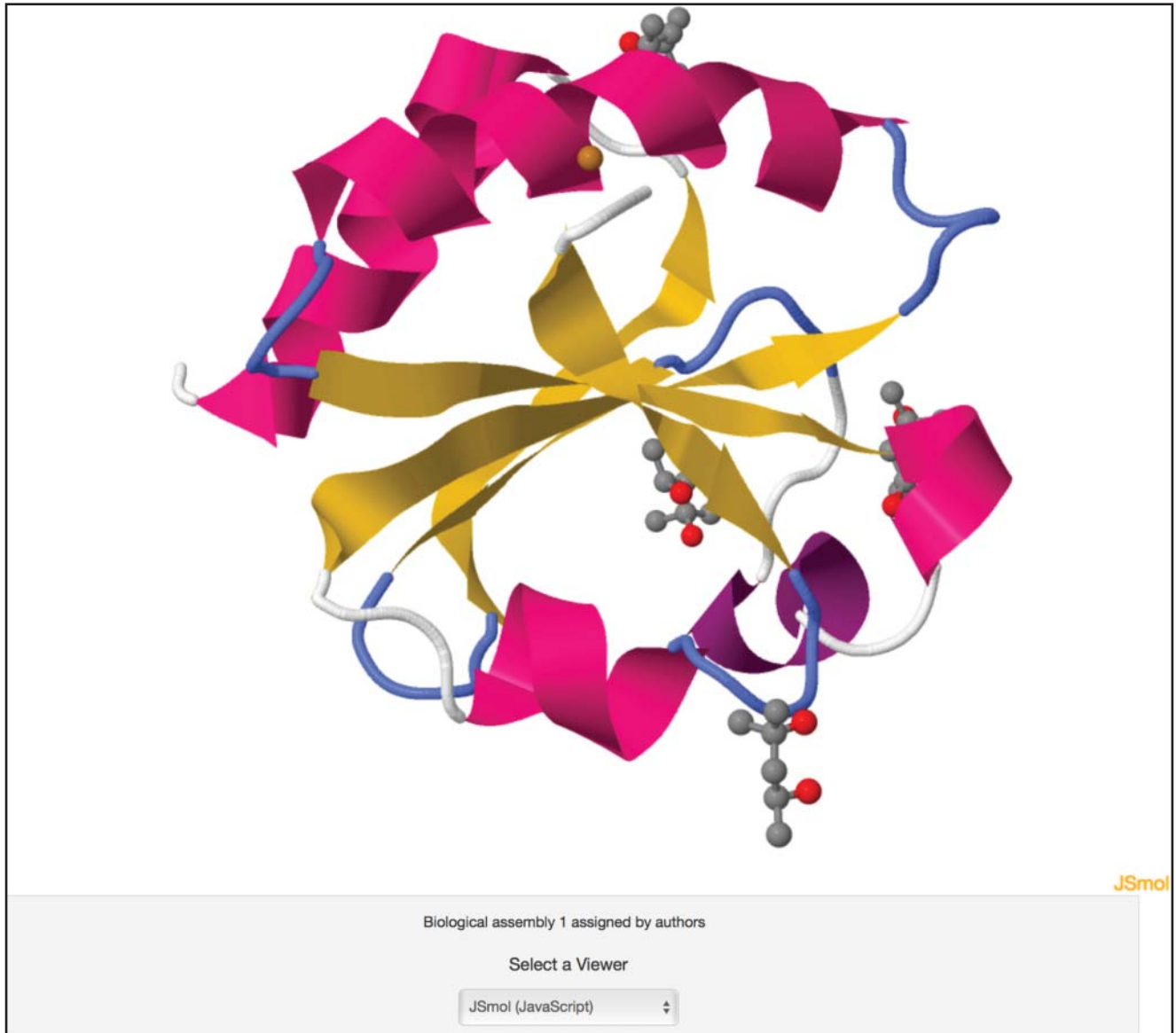


Figure 12.9 A screenshot of an image of *Escherichia coli* thioredoxin as generated by JSmol.

process than finding sequence homologs. Furthermore, structural similarities are somewhat more ambiguous and difficult to describe than sequence similarities. Currently, the PDB provides neighbor assessments using five different packages: FATCAT (Ye and Godzik 2004), Dali (Dietmann et al. 2001), TM-align (Zhang and Skolnick 2005), TopMatch (Sippl and Wiederstein 2008), and CE (Shindyalov and Bourne 2001). Each of these tools provides a slightly different perspective of what is structurally similar to a given protein. While the results provided by all five are almost identical for trivially obvious similarities between structures, they can differ substantially for less obvious cases. Nevertheless, important and unexpected relationships can be found by inspecting these Structure Similarity lists. This is because structure tends to be far more conserved than sequence. In other words, very remote evolutionary relationships can be identified by structure comparison that would otherwise be identifiable via sequence comparison alone. Examples of such unexpected and intriguing relationships include the remarkable similarity between the bacterial toxin colicin A and eukaryotic globins, as well as similarities between the eukaryotic Pituitary Octamer Unc (POU)-specific DNA binding domain and the bacterial lambda repressor (Dietmann et al. 2001).

Other Structure Databases

The PDB is not the only repository of structural data. In fact, there are several secondary or curated structure databases that take the raw data from the PDB and massage or combine it with other data to create some very useful resources. Two of the largest and most useful databases include the Molecular Modeling Database (MMDB) and Proteopedia.

MMDB

The MMDB is the 3D structure database of the National Center for Biotechnology Information (NCBI; Madej et al. 2014). The MMDB is fully integrated into the NCBI database system, with search capabilities across all NCBI databases and direct links to the NCBI Protein Database, the Conserved Domain Database (CDD), and PubChem. The MMDB has a number of useful functions including a specialized sequence-to-structure search function called Cn3D Basic Local Alignment Search Tool (CBLAST), the Inferred Biomolecular Interactions Server (IBIS), the pre-computed Vector Alignment Search Tool (VAST+), structural neighbors, and the Cn3D visualization tool (both a downloadable program version and a JavaScript version; see Chapter 2). Structural information about a given protein can be accessed through the MMDB's Structure Summary page, which displays a still image of the protein structure, a schematic of its protein, nucleotide, and chemical interactions (via IBIS), its CDD links, a direct link to the iCn3D interactive viewer, and a hyperlink to similar structures determined via VAST+. The data stored in the MMDB are uploaded from the PDB daily, checked for exact agreement between coordinate and sequence data, corrected (if necessary) and then mapped to the NCBI's ASN.1 (Abstract Syntax Notation) format. The MMDB is a wonderful example of how freely available structural data from the PDB can be modified or customized to enrich its content for the benefit of all.

Proteopedia

Proteopedia (Hodis et al. 2008) is essentially a Wikipedia for proteins. Proteopedia was originally conceived as a wiki web resource to present protein structure/function information in a user-friendly manner to the broadest possible audience. Each Proteopedia page contains embedded, animated 3D structures (viewable by JSmol) surrounded by descriptive text containing hyperlinks that change the appearance (view, zoom animation, representations, colors, and labels) of the embedded 3D structure image. More than 100 000 Proteopedia pages have been written so far and some of the better annotated entries contain thousands of words (much like a high-quality Wikipedia page) covering the protein's function, its relevance or history, known disease associations, structure or structure highlights, research applications, links to related PDB structures, and an extensive list of references. Clicking on the hyperlinked text embedded in many of these higher quality entries leads to a short, animated "show" that illustrates the concepts explained in the text. Proteopedia's encyclopedic design makes protein structures much more accessible and provides significantly more background or introductory information about specific proteins than what is typically seen in databases like the PDB or MMDB or even in the scientific literature. By adopting a wiki-style approach, Proteopedia is also able to engage the scientific community to write about and share its rich knowledge of specific or important proteins for the benefit of all.

Visualizing Proteins

As discussed in Box 12.2, protein coordinate files are relatively dull looking. They are simply lists of x,y,z coordinates and provide no visual cues as to what the molecule or molecules actually look like. Prior to the advent of computer visualization software, structural biologists

had to build physical models of protein structures from wood, metal, or plastic components to get a “picture” of the structure they had just determined. This obviously proved to be very challenging – and extremely limiting. To get around these problems, early structural biologists developed ingenious methods to help both themselves and others build, visualize, or conveniently understand protein structures. One of the more innovative ideas included the development of something called the Richard’s Box by Frederic M. Richards in 1968. The Richard’s Box was a large optical comparator that allowed crystallographers to rapidly build physical models of protein structures by viewing the stacked plastic sheets of electron density through a half-silvered mirror. Other visualization approaches involved close collaborations with scientific artists (such as Irving Geis and Jane Richardson), who painstakingly drew or painted hundreds of protein structures from the mid-1960s to the late 1970s. Indeed, it was Jane Richardson, an artist and an accomplished X-ray crystallographer, who developed the well-known ribbon diagram approach for visualizing protein backbones and classifying protein folds (Richardson 1981). In the ribbon view, helices are depicted as ribbon-like springs, beta strands are shown as broad, ribbonized arrows, and random coil regions as thin wire or spaghetti-like connectors. The work of both Geis and Richardson has largely inspired the way in which protein structures are visualized on computers today.

By the mid- to late 1970s, computer chips and video terminals became sufficiently fast and sufficiently robust for computers and molecular graphics software to be used for building molecular models of full-sized proteins. The first computer graphics programs for structural biology were FIT (developed by Stan Swanson in 1975) and FRODO (developed by Alwyn Jones in 1977). Other (mostly commercial) packages appeared soon thereafter and computer-based 3D structure visualization became increasingly common and increasingly more sophisticated. Indeed, as can be seen in Figure 12.9, molecular graphics tools allowed structural biologists to create virtual protein models that looked every bit as real as the protein molecule itself. Thanks to computers, proteins could now be colored, shadowed, tinted, textured, and illuminated in thousands of different ways to create very compelling and very impressive molecular art. Molecular graphics software also allowed proteins to be zoomed, rotated, shrunk, and expanded using simple text commands or convenient joysticks. It was during these early days of computer modeling that certain rendering conventions emerged, leading to the establishment of four standard ways to depict proteins: wireframe models, ball-and-stick representations, CPK (Corey, Pauling, and Koltun) or space-filling models, and Richardson ribbon diagrams (also called “cartoon” representations). These conventions are still followed today. Examples of these four rendering styles are shown for the protein ubiquitin (1UBQ) in Figure 12.10.

While computer-based molecular visualization revolutionized structural biology in the 1970s and 1980s, it was only accessible to those who had access to very expensive computers equipped with costly, hard-to-operate software. All of that changed in the 1990s. Starting with the rapid improvements in computer hardware in the late 1980s, the sharp drop in computer costs in the early 1990s, and the source code release of a software package called RasMol (Sayle and Milner-White 1995), molecular graphics became accessible to everyone. Originally introduced in 1993, RasMol (for RASter MOLEcule) represented a major breakthrough in software-driven 3D rendering. Its innovative code design and remarkably fast ray-tracing algorithms made even the slowest desktop computers appear to have ultra-fast graphics engines. RasMol was coded in C and, because the source code was (accidentally) made public, dozens of other visualization packages have since been built based on RasMol’s codebase. Because RasMol did not depend on a system-specific graphics library, it was able to run on just about any platform and operating system. This near-universality made RasMol very popular. In fact, it is estimated that, at its peak, RasMol had more than 1 million users.

With the rapid expansion of the World Wide Web in the late 1990s and the growing desire to view molecules over the web, a number of groups took advantage of RasMol’s freely available source code to make more sophisticated, more user-friendly rendering packages. For instance, in 1997 Molecular Design Limited (MDL) Information Systems converted a large portion of

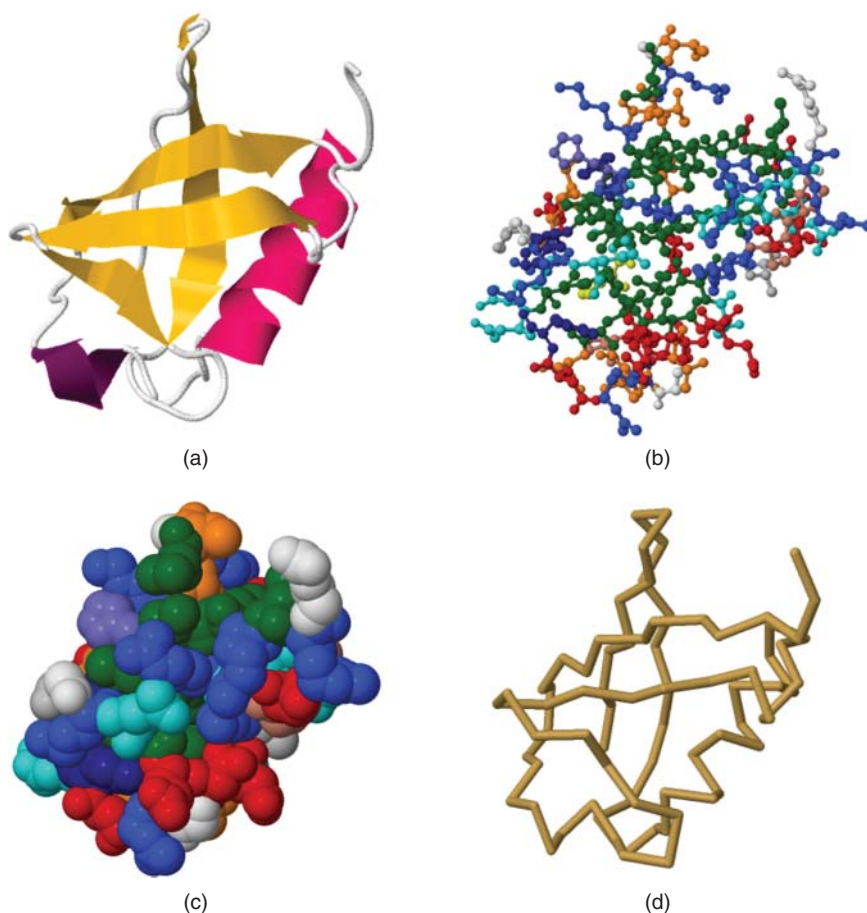


Figure 12.10 An illustration of the four major approaches to rendering protein structures using ubiquitin (1UBQ) as an example. (a) A ribbon diagram that accentuates secondary structure. (b) A ball-and-stick diagram. (c) A space-filling CPK (Corey, Pauling, and Koltun) model. (d) A wire-frame model.

RasMol's C code to C++ and added new functionality including surface rendering and animation to create a browser plug-in called Chime (CHEMical MIME, where MIME stands for Multi-part Internet Mail Extension). Chime was among the first visualization tools to offer an intuitive graphical user interface (GUI) with pull-down menus and mouse click and dragging operations to make the coloring, manipulation, and labeling of protein structures fast and simple.

Other Chime-based packages soon emerged in the early 2000s, including STING Millennium (Higa et al. 2004) and Protein Explorer (Martz 2002). WebMol (Walther 1997) is another example of a package built from RasMol but, in this case, the source code was converted from C to Java. WebMol was the first of many open-source molecular graphics Java applets to appear, along with others such as Jmol (Herráez 2006), Simple Viewer, Protein Workshop, Ligand Explorer, and Kiosk Viewer. Indeed, for the past 15–20 years, Java applets have dominated web-based molecular visualization because of their platform independence, easy installation, and rapid start-up. However, as mentioned above, Java applets have certain security issues and, with the growing concerns over web security, fewer web browsers are supporting Java applets. As a result, the preferred route for molecular visualization over the web is now through JavaScript viewers such as JSmol (Hanson et al. 2013), NGL Viewer (Rose and Hildebrand 2015), PV, and iCn3D (NCBI Resource Coordinators 2017). Nearly all of these viewers support the standard rendering styles (ball-and-stick, CPK, wireframe, and ribbon/cartoon), interactive zooming/rotating, custom coloring, atom/residue picking, labeling, and secondary structure assignment. Furthermore, all of these viewers make use of the Web Graphics Library (WebGL).

WebGL is a JavaScript Application Programming Interface (API) for rendering 3D graphics within any HTML5 compatible web browser without the use of plug-ins. WebGL programs consist of control code written in JavaScript and shader code that is written in OpenGL Shading Language (GLSL), which is all executed on a computer's graphics processing unit (GPU).

While JavaScript viewers have taken the internet by storm, they are not ideal for sophisticated molecular graphics applications. If users wish to create very high-quality or very specialized images or if they need to measure angles, change residues, rotate bonds, calculate energies, or perform other structural manipulations, they must turn to another kind of downloadable 3D visualization software. The two most popular "high-end" protein visualization tools are PyMOL and Swiss-PdbViewer or DeepView (Kaplan and Littlejohn 2001). PyMOL is a downloadable molecular visualization program that is compatible with all major operating systems (MacOS, Windows, and Unix). It is one of just a few open-source model visualization tools available for use in structural biology. "Py" is included in the name because PyMOL is built using the Python programming language. PyMOL is well known for its speed and the very high quality of its molecular visualizations. These visualization features include a wide range of impressive surface-rendering options and ray-tracing capabilities, as well as movie-like animations. Given its ability to generate such high-quality images, nearly 25% of all published images of protein structures in the scientific literature have been generated using PyMOL. In addition to its impressive rendering capabilities, PyMOL also supports a number of structure manipulation tools such as homology modeling, energy minimization, molecular docking, and molecular sculpting. These are available through a variety of freely available plug-ins that have been developed by the PyMOL community. PyMOL handles both text commands and interactive point-and-click operations through a relatively primitive GUI. Owing to PyMOL's enormous popularity, many excellent YouTube tutorials are now available for users wishing to learn more about this program.

DeepView is a freely available, closed-source program originally from Glaxo Wellcome (now GlaxoSmithKline) that is compatible with Windows, MacOS, and Linux operating systems. DeepView supports a range of rendering and modeling capabilities including surface rendering, homology modeling, structure quality (threading) evaluation, energy minimization, site-directed mutagenesis, loop rebuilding, electrostatic field calculation, structure superposition, Ramachandran plot generation, sequence-structure viewing, and the list goes on. With



Figure 12.11 An example of the high-quality images that can be created using advanced ray-tracing methods found in some of the better molecular visualization programs. This image is derived from the coordinates of the hepatitis A virus HAV-3C protease.

DeepView offering so much, it is not exactly the most user-friendly package, especially for novices. Nevertheless, an excellent tutorial prepared by Dr. Gale Rhodes at the University of Southern Maine offers a fine starting point that allows even beginners to learn how to use this superb visualization and modeling package. Several YouTube tutorials are also available. A particularly appealing feature of DeepView is its capacity to export files that are compatible with the freeware ray-tracing package called POV-Ray (Persistence Of Vision – Ray Tracing). POV-Ray allows the more artistically inclined modelers to create stunning images of proteins and protein complexes that are suitable for an art gallery or even a journal cover (Figure 12.11).

While we have only covered a few visualization programs in this chapter, it is important to note that there are now dozens of freely available macromolecular visualization programs that can be found online. Selecting the best one is very much an individual decision, not unlike choosing a computer or buying a cell phone. Ease of use, stability, platform compatibility, and function are all important considerations. Regardless of the program chosen, one should always remember that the central role of visualization software is to create an image that can convey important scientific information in a visually pleasing manner. Taking the time to create a high-quality image and using the right program for the right kind of task can make a tremendous difference to the message one is trying to deliver. Remember, “a picture is worth a thousand words.”

Protein Structure Prediction

Ever since the first protein structure was determined, computational biologists and computational chemists have attempted to develop software that could predict the 3D structure of proteins, using only their sequence as input. Indeed, some of the first bioinformatic programs ever written were directed at trying to solve the “protein folding problem” (Gibson and Scheraga 1967; Chou and Fasman 1974). Even though the field is more than 50 years old, protein structure prediction continues to be an active area of bioinformatic research, with many papers being published on the subject each year. Encouragingly, some progress has been made, and it is now possible to predict or model the 3D structure of proteins using at least three different methods: homology (or comparative) modeling, threading (or fold recognition), and *ab initio* methods. All three methods are fundamentally predictive, meaning that the structures generated are models and are not based on raw experimental data derived from X-ray diffraction, cryo-EM, or NMR experiments. Rather, each of these predictive approaches attempts to build on prior knowledge about protein structure and to extrapolate these principles toward the generation of new structures.

Homology Modeling

Of the three predictive methods that are currently available, the most powerful and accurate approach is homology modeling (Marti-Renom et al. 2000). Homology (or comparative) modeling is a robust technique for “predicting” or generating detailed 3D structures of proteins based on the coordinates of known homologs found in the PDB. In homology modeling, the quality of the model strongly depends on the degree of similarity between the query sequence and the matching database sequence, with proteins sharing the highest degree of similarity being modeled best. As a general rule, the average coordinate agreement between the modeled structure and the actual structure drops by about 0.3 Å for each 10% reduction in sequence identity. Furthermore, homology modeling cannot generally be used for predicting structures of proteins having less than ~30% sequence identity to a target protein already in PDB. However, in certain rare cases, homology modeling can be used to generate a reliable 3D structural model of a protein with much less than 20% sequence identity.

Homology modeling is a multi-step process that makes use of sequence alignment, structure modification, database searches, energy minimization, and structure evaluation to generate a structure. More specifically, homology modeling can be decomposed into five different

steps: (i) aligning the query or unknown protein sequence to the sequence of a known structure; (ii) using the alignment to select and replace backbone segments (usually loops which are contained in a special loop library) that need to be altered due to sequence insertions or deletions; (iii) replacing side chains that have been changed due to the alignment or loop insertion/deletion process; (iv) refining the model using energy minimization to relieve collisions or steric strains; and (v) validating the model using visual inspection and software validation tools. The most critical step to homology modeling is the first step – alignment. An incorrect alignment will have a domino-like effect by increasingly disrupting the remaining steps and eventually leading to a seriously flawed model. To reduce the problems of a single pairwise alignment error, many homology-modeling packages generate alignments from multiple database homologs (if they exist) to improve the reliability of this all-important alignment step.

Originally, homology modeling was a very interactive, manually intensive process that depended critically on the expertise of the user and the availability of specialized 3D visualization software and hardware. Fortunately, many of these complex, time-consuming steps have been automated and now homology modeling can be done by just about anyone on just about any computer. In addition to several high-quality commercial packages, there are a number of excellent freely available homology-modeling packages, including MODELLER (Sali 1998), DeepView, and HHpred (Söding et al. 2005) that can be downloaded and installed on the MacOS, Unix, and Windows platforms. MODELLER is among the oldest (developed in 1989) and is perhaps the best known homology-modeling package. It uses a method called “satisfaction of spatial restraints,” in which a set of geometrical restraints are used to create a probability density function for the location of each atom in the protein. MODELLER needs a sequence alignment between the target amino acid sequence to be modeled and a template protein with a known structure. MODELLER has several variants, including EasyModeller (Kuntal et al. 2010), which provides a user-friendly GUI to MODELLER, and PyMod, a free PyMOL plug-in. Furthermore, millions of MODELLER-generated protein structures are housed in MODELLER’s homology-modeling database, called ModBase (Pieper et al. 2014).

More recently, homology modeling has become available on the web. These web-accessible services include the SWISS-MODEL server (Schwede et al. 2003), the CPHModels server (Nielsen et al. 2010), the ModWeb server (Pieper et al. 2014), the HHpred server (Söding et al. 2005), 3D-JIGSAW (Bates et al. 2001), and PROTEUS2 (Montomerie et al. 2008). Typically, all one has to do is type or paste in the sequence of the protein of interest and press the submit button. A 3D structure will be returned to the user via e-mail within a few minutes to a few hours. HHpred and PROTEUS2 are known to be particularly fast, with response times of a few minutes. An example of a homology model generated for *Escherichia coli* thioredoxin generated from a template with just 26% sequence identity is shown in Figure 12.12.

Most published homology-modeling programs and servers are rigorously tested, so the results from any given package or web server are actually quite trustworthy. Many packages have been assessed through the Critical Assessment of Protein Structure Prediction (CASP) evaluation process. CASP is a community-driven initiative that has been conducted every 2 years since 1994. The purpose of CASP is to provide an independent, unbiased or “blind” assessment of different programs or methods in protein structure prediction, including homology modeling, threading, and ab initio prediction. The organizers of CASP work with X-ray crystallographers and NMR spectroscopists who provide coordinates of several dozen newly determined or about-to-be-determined protein structures. The sequences for these structures are then sent to registered CASP predictors who typically have several months to generate structures and deposit their predictions with the CASP organizers. Once the competition closes, all the submitted structures are evaluated using a variety of rigorous structure comparison techniques (described in Protein Structure Comparison). Based on CASP and other independent evaluations, MODELLER, SWISS-MODEL, and 3D-JIGSAW appear to provide the best performance among homology-modeling servers. Overall, homology modeling is the most reliable, most accurate, and most widely used method for protein structure prediction. With the enormous size of the PDB (now >120 000 structures) and its

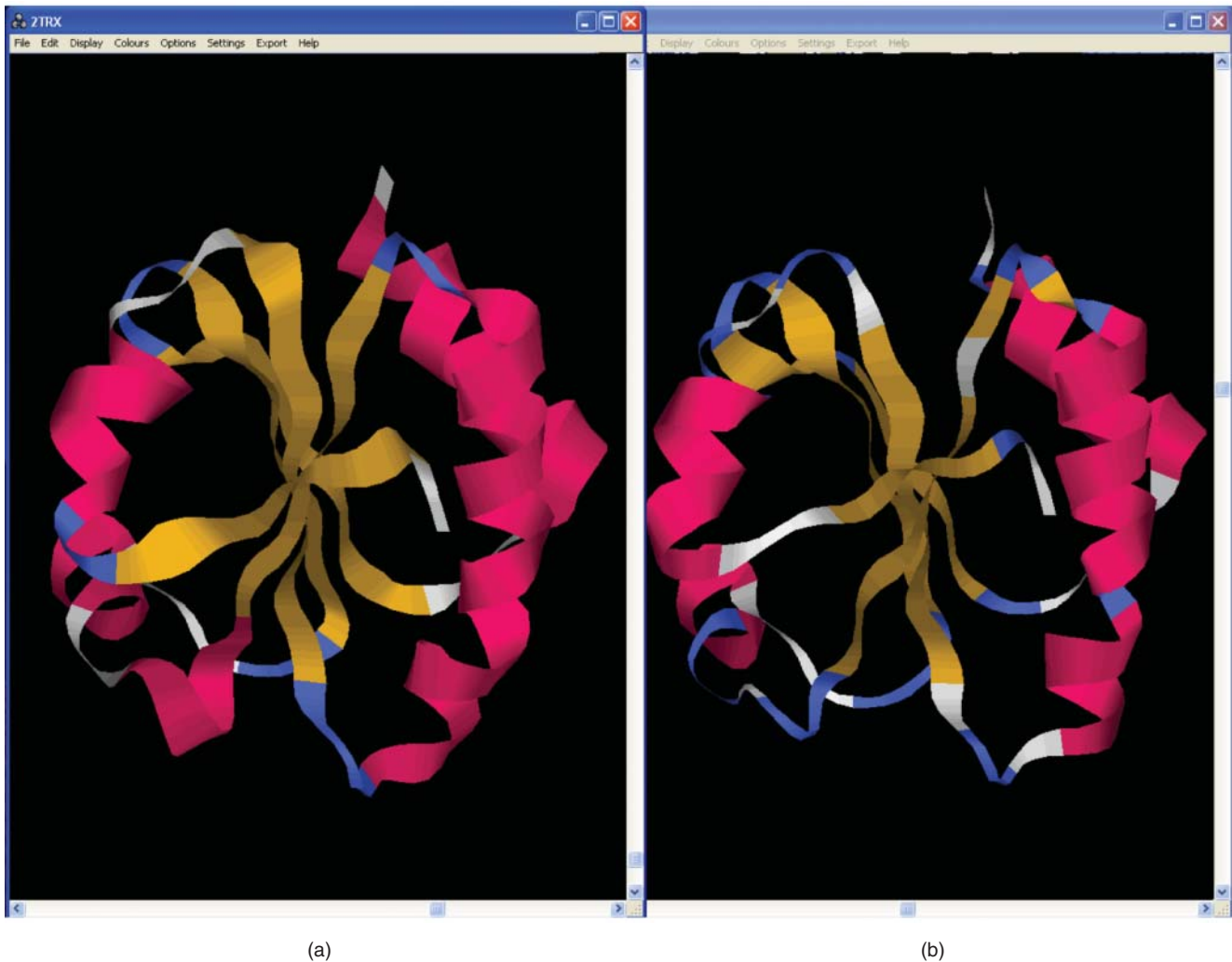


Figure 12.12 An illustration of a homology model (b) of *Escherichia coli* thioredoxin generated using human thioredoxin (3TRX – 26% sequence identity) as a template. The actual X-ray structure of *E. coli* thioredoxin is shown in (a). Note the very good overall similarity by visual inspection.

comprehensive coverage of most known folds, almost any given protein sequence has a very good chance of being successfully generated via homology modeling.

Threading

Threading (or fold recognition) is a method for predicting the structure of or recognizing a common fold in proteins having essentially no sequence homology to any protein in the PDB (Bowie et al. 1991; Bryant and Lawrence 1993). In other words, threading is a structure prediction technique that picks up where homology modeling leaves off. Unlike homology modeling, which strives for accurate models, threading is limited to generating more approximate models or approximate folds. Threading received its name because it superficially resembles the method used to thread a thin tube down or through a plumbing pipe system. In the course of threading the tube or probe (called a “snake”) through the pipe, the wire takes on the shape of the surrounding pipe (Figure 12.13). If we view the backbone structure of a protein as being very similar to a highly contorted hollow pipe (like an elaborate plumbing system), we could ask what would happen if we threaded a completely different protein sequence through this backbone pipe. Intuitively, we would expect that if the probe sequence resembled the sequence belonging to the original pipe, then the fit would be rather good, with the amino acid side

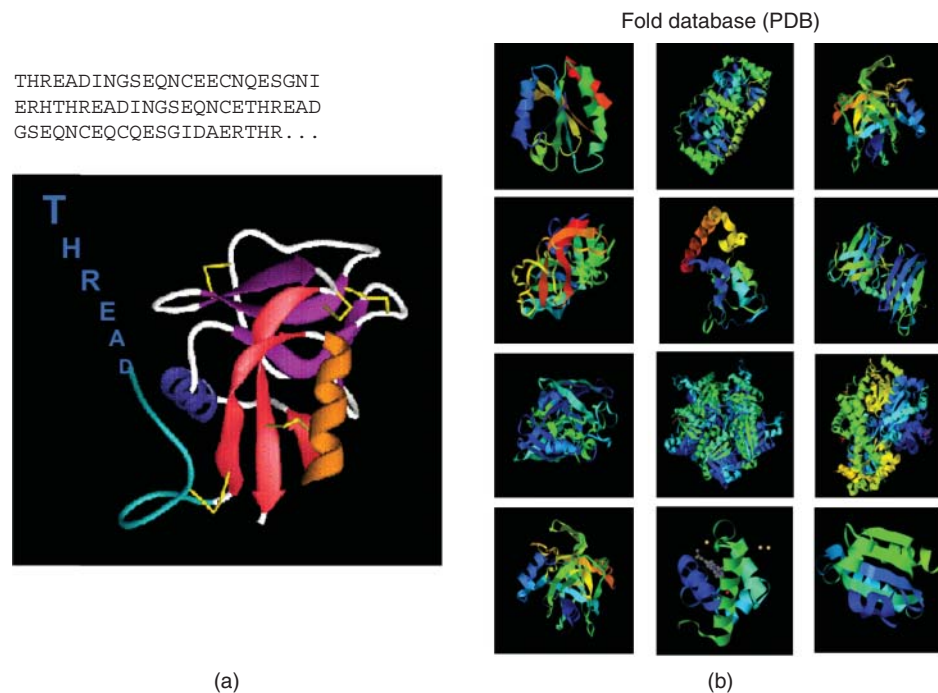


Figure 12.13 A schematic illustration of how threading is performed. (a) A query protein with the sequence THREADINGSEQ... is passed through, one residue at a time, the three-dimensional structure of each protein in a fold database (shown by the structures in b). The energy or quality of the fit is evaluated each time, with the highest scoring match being the most likely fold.

chains packing closely against one another. If, on the other hand, the probe sequence was very different than the pipe sequence, then we might find that when the probe sequence is finally fed through the pipe it would fit rather poorly, with side chains smashing into each other or pointing in the wrong direction.

If one were to take this threading procedure one step further and automate the process, then it would be possible to run hundreds or thousands of different probe sequences through this protein backbone pipe, one at a time. As each sequence is fed through, the fit is evaluated to determine which sequence fits best with the given template pipe or backbone fold. This evaluation may be done quickly using some empirical energy term or some measure of packing efficiency. In this way, it is possible to assess which protein sequences are compatible with the given backbone fold. Clearly one would expect that those sequences that are highly homologous to the original template sequence should fit best. However, it has also been found that this simple-minded approach can occasionally reveal that some completely unrelated sequences can also fit into this fold. When these kinds of sequences are discovered, one is, in effect, predicting the tertiary fold of an unknown protein (i.e. performing a 3D structure prediction).

Three-dimensional structures or folds predicted from threading techniques are not generally of high quality, with a typical RMSD between the correct structure and the modeled structure being $>2 \text{ \AA}$. However, threading methods can and do reveal the approximate shape and overall fold of proteins that seem to have no known structural homologs. Threading rose to prominence in the early 1990s when it was used to model the approximate structure of leptin, a protein that plays an important role in obesity. At the time, no sequence or structural homolog was known and all attempts at homology modeling had failed (Madej et al. 1995). The threading model generated by Madej et al. suggested a general mechanism for the protein's activity that was later found to be quite accurate. Since then, threading has become a real darling of the protein structure prediction community.

Given the popularity of threading, there are now a large number of web-based threading services available, including Phyre2 (Kelley et al. 2015), HHpred (Söding et al. 2005), RaptorX

(Källberg et al. 2014), LOOPP (Vallat et al. 2009), and MUSTER (Wu and Zhang 2008). More recently, the use of multiple threading servers (so-called meta-servers) that combine the results of several threading predictions appear to be yielding the best results for structure prediction. Examples of meta-threading servers include eThread (Brylinski and Lingam 2012) and LOMETS (for Local Meta-Threading Server; Wu and Zhang 2007). LOMETS has been integrated into the structure-function prediction tool known as I-TASSER (for Iterative Threading ASSEMBLY Refinement; Yang and Zhang 2015). I-TASSER, which is commonly referred to as the “Zhang server,” as it was developed by Yang Zhang at the University of Michigan, has been consistently ranked as the top server for protein structure prediction in the CASP7, CASP8, CASP9, CASP10, CASP11, and CASP12 competitions (from 2006 to 2016).

Ab Initio Structure Prediction

Ab initio prediction literally means “predicting from the beginning.” In other words, this approach attempts to predict protein structures without prior knowledge of any related 3D structure. Ab initio prediction is generally aimed at identifying new folds, or folds for which there is no sequence similarity whatsoever to existing structures. Over the past decade, significant progress has been made in ab initio protein structure prediction, with smaller (<150 residues) proteins having their structures accurately predicted with surprising regularity. Much of the progress has been due to the work of Dr. David Baker and his group at the University of Washington. In the early 2000s, the Baker group developed a program known as Rosetta (Bonneau et al. 2001). Rosetta uses a large library of peptide fragments taken from known protein structures, along with a specially developed Monte Carlo sampling technique and an intelligent energy function to “fold” proteins (i.e. predict a protein structure). Rosetta does not use homology modeling, threading, or template-assisted structure generation and, as such, represents a true de novo or ab initio approach to predicting protein structures. Rosetta was remarkably successful in the early CASP competitions for ab initio structure prediction. Using the same search concepts and intelligent energy functions, Rosetta has evolved into a number of other variants, including RosettaDock (for docking proteins together), RosettaDesign (for designing novel proteins), and RosettaLigand (for docking small molecules to proteins). Many of these variants are now freely downloadable through the RosettaCommons web site. Additionally, several of the Rosetta programs are now freely available as web servers including ROSIE (Lyskov et al. 2013), Robetta (Kim et al. 2004), and RosettaDesign (Liu and Kuhlman 2006). Rosetta is even available as a distributed “mini-platform” for home computer-based or crowd-sourced protein structure prediction and docking (through Rosetta@home and Foldit).

Rosetta’s success has inspired many others in the protein structure prediction community, and some of the most successful structure prediction programs today (such as I-TASSER) use algorithmic concepts borrowed from Rosetta. However, other approaches to ab initio protein structure prediction also exist. One of the most intriguing makes use of massively parallel molecular dynamics (MD) simulations conducted with custom-built supercomputers containing specially designed, MD-optimized computer chips (Klepeis et al. 2009). These MD simulations have been shown to be sufficiently detailed and sufficiently accurate to model the correct folding of small, fast-folding proteins (Lindorff-Larsen et al. 2011). This is a truly impressive achievement. Another fascinating approach to ab initio or de novo protein structure prediction employs a technique called co-evolutionary coupling (Marks et al. 2011). In this elegant method, multiple sequence alignment (see Chapter 8) is used to infer pairwise residue couplings or spatial interactions via evolutionary constraints. That is, pairs of residues that are far apart in sequence will change in a coordinated way if they are close together in space. For instance, a small residue (say glycine) packed next to a large residue (say tryptophan) can only be substituted with a medium-size residue (say leucine) if the large residue is simultaneously substituted with another medium-size residue (say valine). These coordinated residue mutations or “couplings,” as inferred through the sequence alignment and appropriate statistical analyses, are then used to create pairwise atomic constraints. These pairwise constraints can

then be used to construct atomic-resolution structures. This co-evolutionary coupling method only uses sequence data as input (no homology modeling is done) and it has been shown to generate protein structure models that are within 3–5 Å RMSD of the experimentally determined structures (Marks et al. 2011).

While the progress being made in *ab initio* structure prediction is quite impressive and a solution to the protein folding problem via computing seems to be almost at hand, it looks like much of this elegant *ab initio* work may be for naught. Indeed, thanks to the enormous efforts of structural biologists over the past 50 years, it appears that most of the naturally possible protein folds are now known. Indeed, the number of known protein folds grew from 405 in 1997, to 1086 in 2007 (Levitt 2007) to just 1228 in 2017, with a near-complete absence of new folds being identified over the past several years. This means that almost every protein structure that is solved by NMR, X-ray crystallography, or cryo-EM today is quite similar to one or more already existing structures in the PDB. Therefore, it is now possible for almost anyone to use freely available homology modeling or freely available threading web servers to figure out the structure of almost any known protein directly from its amino acid sequence. In other words, the protein folding problem has essentially been solved by “brute force.”

Of course, this does not discount the need for continuing to develop better prediction software, nor does it negate the need for structural biologists or structural biology. There will always be plenty of questions regarding protein–protein interactions, protein dynamics, protein energetics, and protein–ligand binding that will need to be solved by careful measurement, exacting simulation, and well-designed experimentation. Likewise, with the growing realization that up to 30% of all proteins or protein domains are actually unstructured or intrinsically disordered, a host of new structural challenges are now facing structural biologists, computational biologists, and database curators (Varadi et al. 2014).

Protein Structure Evaluation

Whether the coordinates for a protein structure have been obtained experimentally (using NMR, X-ray, or cryo-EM) or by modeling (through homology or threading), it is always important to ask this very simple question: “How good is this structure?” A poor structure, like a poor model, can lead to misinterpretation of how a protein works, how it is related to other proteins, or where a potential ligand may or may not bind. On the other hand, a high-quality structure can reveal a tremendous amount of biologically important information and can serve as the basis to test new hypotheses on folding or function, to design and construct mutants, or to design new drugs. A large majority of the experimentally determined structures in the PDB are really quite excellent and, certainly, most structural biologists strive to generate the best structures they can. However, there are at least a dozen examples of protein structures in the PDB that have been found to be so seriously flawed that they had to be withdrawn (Hooft et al. 1996). There are also dozens of protein structures that are poorly resolved (>3 Å resolution), have mislabeled residues or atoms, are missing lengthy tracts of sequence, or provide only α coordinates. With the advent of NMR spectroscopy as an alternative to X-ray crystallography, we are now seeing that many protein structures or parts of protein structures actually differ quite substantially between solution and solid (crystal) state conditions. Even among different crystal forms of the same protein, it is quite normal to see an average difference of ± 0.5 Å in atomic displacement or $\pm 7^\circ$ in backbone dihedral angle variation. These structural variations are not restricted to experimentally determined structures. For instance, homology models invariably exhibit differences between themselves and the real structure (once determined), with the extent of the differences increasing by about 0.3 Å for each 10% drop in sequence identity. In addition, homology models are frequently found to have at least one or two regions that are modeled incorrectly, because of sequence alignment errors, loop insertion errors, or energy refinement errors. While these comments may seem to cast doubt on the reliability and utility of many protein structures, their intent is primarily to inject an appropriate degree of

caution or skepticism with which all scientific data should be treated. These comments are also intended to underline the importance of always trying to answer the question we began with: “How good is this protein structure?”

Protein structures are remarkably complex and highly variable. This complexity makes it almost impossible to simply look at a protein structure and assess its quality or correctness. However, by studying large numbers of protein structures and by focusing on those structures that exhibit particularly good resolution, structural biologists have realized that there are some near-universal characteristics to high-quality structures. In particular, when considering the structures of water-soluble proteins, good protein structures should:

- minimize the number of torsion angles in disallowed regions of the Ramachandran plot
- maximize the number of hydrogen bonds
- minimize the number of exposed hydrophobic residues
- maximize the number of exposed polar or charged residues
- minimize the number of interstitial cavities or packing defects
- minimize the number of number of non-bonded atoms within 2.6 Å
- minimize the standard deviation in hydrogen bond energies
- minimize the standard deviation in dihedral angles for helices
- have a low *R* factor (<0.20 for X-ray structures) or a low backbone RMSD value (<0.8 Å for NMR structure ensembles).

Some of these characteristics or features also appear to represent underlying rules of protein folding. Therefore, it is not surprising that they should be reiterated in the structural features of most proteins. Interestingly, many of these characteristics can also be quantified or measured directly from protein coordinate data. These observations have led to the development of a number of excellent software programs for automatically evaluating protein structures and protein models, including the Dictionary of Secondary Structure for Proteins (DSSP; Kabsch and Sander 1983), PROCHECK (Laskowski et al. 1993), the Volume, Area, Dihedral Angle Reporter (VADAR; Willard et al. 2003), and MolProbity (Davis et al. 2007).

DSSP is an open source program, written in C++, designed to produce a compact, sequence-centric summary of local protein structure features (Kabsch and Sander 1983). It is also available as a web server. DSSP uses a very stringent method to identify hydrogen bonds and hydrogen bonding patterns, which in turn are used to identify and label seven different kinds of secondary structures: alpha helices (H), 3/10 helices (G), pi helices (I), beta bridges (B), extended beta strands (E), hydrogen-bonded turns (T), and bends (S). DSSP’s definition of secondary structure has become the de facto standard for secondary structure annotation in the PDB and the reference set for most secondary structure prediction schemes. In addition to performing automated secondary structure identification and assignment, DSSP also determines the ASA of individual residues using the ANAREA algorithm. The results are presented in a simple digital scale (0–9), with 0 corresponding to fully buried and 9 being fully exposed.

PROCHECK was, perhaps, the first quantitative protein structure evaluation program, and is still one of the best available (Laskowski et al. 1993). PROCHECK is a downloadable program that accepts PDB-formatted X-ray coordinate files as input and uses DSSP to identify secondary structure and calculate ASA. It also calculates torsion angles (backbone and side chain), bond angles, interatomic distances, and other relevant structural properties. By comparing these values with those observed for very high-resolution or high-quality structures, PROCHECK is able to provide an estimate of the quality or equivalent resolution for any given query structure. One of PROCHECK’s most appealing features is its colorful graphical reports that are automatically generated (including Ramachandran plots, secondary structure markups, and scatter plots) along with tables, explanations, and references (Figure 12.14). Inspection of these graphs or tables allows users to quickly identify problem areas or zero-in on suspicious and unusual structural features.

VADAR is a protein structure evaluation web server that assesses both NMR and X-ray structures using PDB coordinates or PDB ID codes as input (Willard et al. 2003). Like the

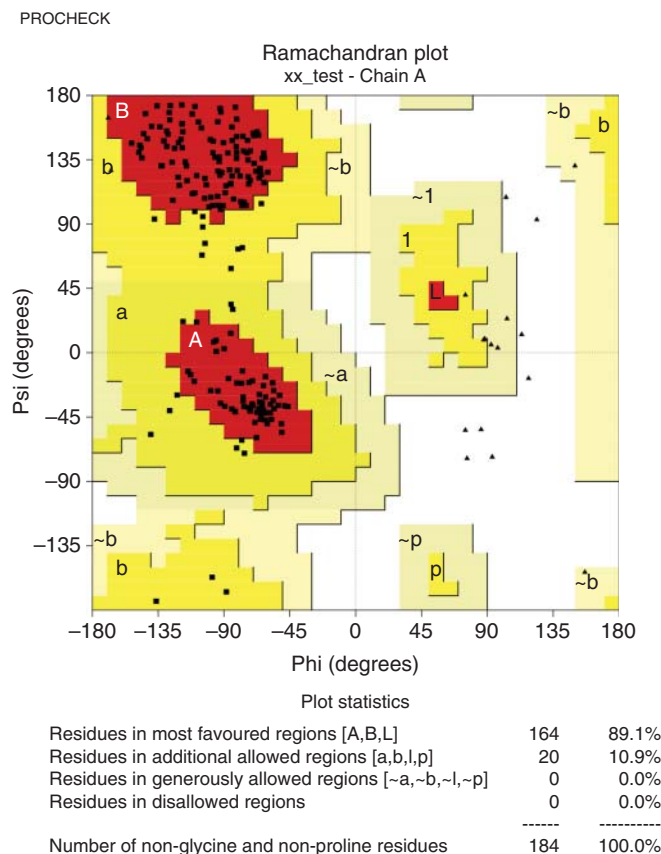


Figure 12.14 An example of the high-quality postscript output data from PROCHECK.

other programs already mentioned, VADAR measures dihedral angles, identifies hydrogen bonds, and measures interatomic distances to help evaluate protein structures. Unlike the other programs, however, VADAR uses a more comprehensive approach to identifying secondary structures in which three methods are used to generate a consensus secondary structure. It also identifies and classifies beta turns, identifies side chain hydrogen bonds or salt bridges, calculates packing volume (in \AA^3), determines exact ASA (in \AA^2), performs packing “defect” checks and buried charge evaluation, calculates threading and surface free energies, determines residue disposition, and compares many of these values with those that would be expected among high-quality structures. A variety of tables are generated for different parts of the protein (main chain and side chain), as well as a summary table describing and identifying suspicious features found in the protein. Ramachandran plots (with outliers marked) and structure quality graphs (JPG or PNG) are also automatically created.

MolProbity represents a newer generation of structure evaluation web server that uses all atom contact analysis to assess protein structures (Davis et al. 2007). In particular, MolProbity adds and optimizes the geometry of hydrogen atoms (using a program called REDUCE) to all input structures and then calculates their H-bond, steric clash, and van der Waals contacts. This kind of contact analysis is remarkably sensitive because hydrogen atoms are not only the most abundant atoms in proteins, they also make the most of the atomic contacts. As a result, contact deviations detected at the hydrogen atom level will amplify and identify problems with any attached carbon, oxygen, and nitrogen atoms, including their bond lengths, placement, and angles. Like VADAR, MolProbity is capable of processing both NMR and X-ray structures and, also like VADAR, it produces Ramachandran, amino acid rotamer, and covalent geometry measures to help with general structure assessment. MolProbity is a very powerful structure assessment tool and its output is now a part of the wwPDB Validation Report provided for all protein structures in the PDB.

Protein Structure Comparison

Similar to sequence comparison, structure comparison lies at the heart of structural bioinformatics. In the same way that sequence comparisons can provide tremendous insight into the origins, function, location, interactions, and activity of a protein, so too can structure comparison. In fact, because structure is actually much more conserved than sequence, structure comparisons allow us to look even further back into Earth's prehistory to track the origins and evolution of many key enzymes and proteins. Unfortunately, structure comparison is a much more computationally difficult process than sequence comparison. In sequence comparison, it is possible to use character string matching or dynamic programming methods to easily and rapidly generate alignments and to identify regions of sequence similarity. In structure comparison, a completely different scheme must be used, as one is comparing or aligning complex 3D shapes rather than simple two-dimensional character strings. While computers are naturally very good at handling strings, they are not particularly good at identifying or comparing 3D objects. Indeed, humans still outperform even the fastest computer in recognizing or comparing modestly dissimilar 3D objects.

Nevertheless, there are tools and techniques that make it possible to compare near-identical or relatively similar 3D structures. The most common method is called structure superposition. Superposition or superimposition is simply the process of rotating or orienting an object until it can be superimposed on top of a similar object. It is not unlike the process humans normally perform when putting the last piece of a jigsaw puzzle into place, rotating, and translating the puzzle piece around until it finally fits. The simplest route to 3D superposition is to identify a minimum of two sets of three common reference points, one set for the object to be superimposed and another set on the reference object that is being overlaid. Once these points are identified, the object to be superimposed can be rotated and translated until the two sets of reference points are almost matching (i.e. minimally different). The problem, of course, is knowing which three reference points are most appropriate. Humans are very good at this, computers are not. The problem with proteins is further complicated because we typically want to superimpose not just three points but literally hundreds of points (or atoms) at the same time.

Fortunately, there are mathematical approaches that allow this superposition process to be done, as long as the reference points are identified and as long as the two objects have the same number of identified points. These approaches include Lagrangian multipliers, quaternion methods, and matrix diagonalization techniques. It is beyond the scope of this chapter to explain the details of these methods, but suffice it to say that all these approaches are very fast, mathematically robust, and a number of them have been coded into readily available computer programs. It is possible to use the same techniques to superimpose more than two structures, as is frequently done for NMR structure ensembles. In this case, an iterative approach is taken where the first two superimposed structures are averaged to create a single structure, which is then used as a template to superimpose the third structure. The process of averaging and adding is repeated until all the structures have been superimposed. Typically, the two most similar structures are superimposed first, with the least similar structure being superimposed last, much as is done in a progressive multiple sequence alignment.

A number of structure visualization programs such as PyMOL, Jmol (Herráez 2006), and DeepView (Kaplan and Littlejohn 2001) are particularly good at both performing and visualizing molecular superpositions. There are also a large number of web servers that perform molecular superpositions of pairs of protein structures. Some of the more popular ones include SuperPose (Maiti et al. 2004), FATCAT (Ye and Godzik 2004), CE (Shindyalov and Bourne 2001), and TM-align (Zhang and Skolnick 2005). Nearly all of these servers allow users to upload a pair of PDB IDs or a pair of PDB files and to simply press the submit button to generate coordinate data. Some of the servers, such as SuperPose, allow users to superimpose more than two structures. Certain servers (SuperPose, CE, and TM-align) perform rigid superposition while others (such as FATCAT) perform a more flexible superposition. The output for these

web servers are simple PDB coordinate lists that can be viewed by any number of visualization tools, image files of the superpositions, as well as information on the alignment, number of equivalent residues, RMSDs, or alignment scores.

The establishment of methods and criteria to quantitatively compare protein structures (i.e. structure superposition) led to the establishment of a number of databases containing common protein folds. This is the equivalent of grouping sequence families together to identify common sequence motifs, as is done with the Pfam, PROSITE, and InterPro databases (see Chapter 7). The equivalent databases, in terms of structure, are CATH (Pearl et al. 2000) and Structural Classification of Proteins (SCOP) (Murzin et al. 1995). Using these kinds of databases, it is possible to discover unexpected or undiscovered relationships between distantly relative proteins or to find fascinating examples of convergent structural evolution. CATH (which stands for Class, Architecture, Topology, Homology) is a database that groups proteins into a taxonomy based on their secondary structure content, fold, and sequence similarity. The result is a hierarchical domain classification schema that allows protein structures to be logically grouped and compared. CATH entries are derived from higher resolution protein structures ($<3.0 \text{ \AA}$) in the PDB, with multidomain proteins being partitioned into their constituent domains prior to classification. At the top of the hierarchy is the Class level, which is determined automatically by the secondary structure content. There are three broad classes, mainly alpha, mainly beta, and alpha/beta (see above). At the Architecture level, protein structures are further divided according to the overall domain shape and orientation of the secondary structures. This is done manually using naming conventions found in the literature. Third in the hierarchy is the Topology level, where common architectures can be further divided into groups according to their secondary structure connectivity and general shape. At the lowest level of the hierarchy, proteins are grouped according to their sequence identity ($>35\%$) and length of sequence match ($>60\%$). The CATH database can be searched by text, identifier, protein sequence, or by PDB structure (Figure 12.15). CATH is also linked to the Gene3D database that contains the predicted CATH domains for tens of millions of protein sequences derived from public databases.

The SCOP database is a similar hierarchically structured database providing a slightly different taxonomic partitioning. Like CATH, the SCOP database aims to provide a comprehensive description of the structural and evolutionary relationships between essentially all protein structures in the PDB. Unlike CATH, the SCOP database is primarily constructed through visual comparison and manual grouping. This process is aided (but not guided by) a number of computational tools. SCOP uses a six-part hierarchy (Species, Protein, Family, Superfamily, Folds, and Class). Species corresponds to a distinct protein sequence from a distinct biological species, Protein corresponds to similar sequences of essentially the same functions that either originate from different biological species or represent different isoforms within the same species, Family corresponds to proteins with similar sequences but distinct functions, Superfamily links together protein families with common functional and structural features inferred to be from a common evolutionary ancestor, Folds corresponds to proteins that have the same major secondary structures in the same arrangement and with the same topological connection, and Class corresponds to proteins with similar secondary structure content and organization. SCOP has seven “true” classes (with four additional specialty classes for non-conforming proteins) that are based on secondary structure content and size. Since 2009, the SCOP database has been evolving into an extended version (called SCOPe; Chandonia et al. 2017) and to a newly updated hierarchy called SCOP2. Both the original SCOP database and the new SCOPe database can be easily browsed, progressing down the hierarchy from Classes to Folds to Superfamilies and so on through hyperlinks, or searched by keywords. SCOPe is much more up to date than SCOP and is full of hyperlinked documents and thumbnail structural images that allow the facile navigation and exploration of structural and evolutionary relationships.

As the time and expense of manually classifying protein structures (as done by CATH and SCOP) is quite considerable, there has been a steady shift toward more automated, less hierarchical approaches. In particular, a number of web-based services have appeared that now allow users to compare newly determined structures against all existing structures in

CATH Domain 2trxA00

Home / Superfamily 3.40.30.10 / Domain 2trxA00

DOMAIN LINKS

- Summary
- Structure
- Sequence
- Neighbourhood

CATH Classification

Level	CATH Code	Description
3		Alpha Beta
	3.40	3-Layer($\alpha\beta\alpha$) Sandwich
	3.40.30	Glutaredoxin
	3.40.30.10	Glutaredoxin

CATH Clusters

Superfamily: Glutaredoxin

Functional Family: Thiol-disulfide isomerase and thioredoxin

Enzyme Information

1.-.-.- Oxidoreductases.
based on mapping to UniProt P0AA25

UniProtKB Entries (1)

P0AA25 THIO_ECOLI
Escherichia coli K-12
Thioredoxin-1

PDB Structure

PDB: 2TRX

External Links:

- PDBSum
- Proteopedia

Method: X-RAY DIFFRACTION

Organism

Primary Citation: Crystal structure of thioredoxin from Escherichia coli at 1.68 Å resolution. Katti, S.K., LeMaster, D.M., Eklund, H. J.Mol.Biol.

Figure 12.15 An example of the CATH database description of *Escherichia coli* thioredoxin, indicating its class (Alpha Beta), architecture (3-Layer ($\alpha\beta\alpha$) Sandwich), topology (Glutaredoxin), and homology (Glutaredoxin) to other related structures.

the PDB. These structure similarity search servers include FATCAT (Ye and Godzik 2004), Dali (Dietmann et al. 2001), TopSearch (Wiederstein et al. 2014), and PDBeFOLD, which was formerly known as SSM (Krissinel and Henrick 2004). Structure similarity search servers are capable of performing a pairwise structure superposition of an uploaded query structure against every structure in the PDB. In this regard, FATCAT, Dali, TopSearch, and PDBeFOLD are the structural biologist's equivalent to BLAST. However, unlike BLAST, the quality of a structural match is not measured by an *E*-value. Rather, these structure comparison algorithms typically report a wide range of "alternative" assessments such as *p*-values, *Z*-values, sequence coverage, rankings, raw scores, and RMSDs. (Note that it is always best to read the output legends carefully, as there is no consensus on scoring methods.) Most servers will also report percent sequence identity over the structurally aligned regions. Smaller *p*- or *Z*-values, smaller RMSDs, higher scores, and higher sequence coverage are all good indicators of structural similarity. The multiplicity of scoring schemes has to do with the fact that these structure similarity search servers perform very sophisticated "alignments" that are far more complicated than those generated by BLAST, incorporating gaps of almost any length, chain reversals, geometrical distortions, and altered topological connectivity of aligned segments.

Many X-ray crystallographers and NMR spectroscopists use FATCAT, Dali, TopSearch, and PDBeFOLD to ascertain if their newly determined structure (that is not yet deposited in the PDB) is a representative of a new fold or belongs to an existing fold. Such a determination can have profound implications for understanding the function and origin of a protein. If a structure of unknown function exhibits significant structural similarity to a structure of known

function, it is often possible to make an assertion about the unknown protein's function. As always, it is particularly interesting and informative to identify those structures where the RMSD is $<2 \text{ \AA}$, yet the sequence identity is $<15\%$. These are examples of either very ancient homologs or potentially interesting cases of convergent evolution.

Of course, not everyone is a structural biologist and not everyone has access to the coordinates of a completely new protein structure, so novel structure similarity search queries are relatively rare. More often, users are simply interested in better understanding an existing structure – either its evolution or its potential function. In these situations, it is much easier (and substantially faster) to upload an existing PDB identifier (rather than a PDB coordinate file) and to perform a search against a pre-calculated database of structural neighbors. These kinds of pre-calculated neighbor searches are supported by VAST+ (Madej et al. 2014), FATCAT, Dali, TopSearch, and PDBeFOLD. They are also available through the PDB Structure Similarity pages that were introduced earlier in this chapter.

Summary

Many of the concepts and ideas we use in bioinformatics today, such as sequence comparison, structure/sequence visualization, structure prediction, electronic databases, and evolutionary analysis can trace their origins to structural biology and the structural biologists who developed many of the earliest bioinformatic tools. Without these important contributions from structural biology and structural biologists, bioinformatics would not be what it is today. More recently, the tables have begun to turn, with structural biologists now looking to bioinformaticians to help solve emerging problems in pattern finding, remote structure comparison, and large-scale distributed data management. This give and take between structural biologists and bioinformaticians is vital to sustaining both fields, and this exchange of expertise and insight will undoubtedly continue for some time to come. Hopefully, this chapter has illustrated how at least some of these interactions have evolved and how structural bioinformatics continues to be integral to gaining a detailed understanding of the engines of life – proteins and enzymes.

Internet Resources

BioMagResBank	www.bmrb.wisc.edu
CASP	predictioncenter.org
CATH/Gene3D	www.cathdb.info
CE	source.rcsb.org/jfatcatserver/ceHome.jsp
CPHModels	www.cbs.dtu.dk/services/CPHmodels
Dali	ekhidna2.biocenter.helsinki.fi/dali/
DeepView	spdbv.vital-it.ch
DSSP	www.cmbi.ru.nl/dssp.html
FATCAT	fatcat.sanfordburnham.org
HHpred	toolkit.tuebingen.mpg.de/#/tools/hhpred
iCn3D	www.ncbi.nlm.nih.gov/Structure/icn3d/full.html
I-TASSER	zhanglab.ccmb.med.umich.edu/I-TASSER/
Jmol	jmol.sourceforge.net
JSmol	jmol.sourceforge.net
LOMETS	zhanglab.ccmb.med.umich.edu/LOMETS
LOOPP	cbsu.tc.cornell.edu/software/loopp
MMDB	www.ncbi.nlm.nih.gov/Structure/MMDB/mmdb.shtml
MODELLER	salilab.org/modeller
ModWeb	modbase.compbio.ucsf.edu/modweb
MolProbity	molprobity.biochem.duke.edu

MUSTER	zhanglab.ccmb.med.umich.edu/MUSTER
NGL Viewer	proteininformatics.charite.de/ngl/html/ngl.html
PANAV	panav.wishartlab.com
PDBe	www.ebi.ac.uk/pdbe
PDBeFOLD	www.ebi.ac.uk/msd-srv/ssm
PDBj	pdbj.org
Phyre2	www.sbg.bio.ic.ac.uk/~phyre2/html/page.cgi?id=index
Proteopedia	proteopedia.org/wiki/index.php/Main_Page
PROTEUS2	www.proteus2.ca/proteus2
PyMOL	www.pymol.org
RaptorX	raptorx.uchicago.edu
RasMol	www.openrasmol.org
RCSB-PDB	www.rcsb.org/pdb/home/home.do
Robetta	robeta.bakerlab.org
Rosetta@home	boinc.bakerlab.org
RosettaCommons	www.rosettacommons.org
RosettaDesign	rosettadesign.med.unc.edu
ROSIE	rosie.rosettacommons.org
SCOP	scop.mrc-lmb.cam.ac.uk/scop
SCOPE	scop.berkeley.edu
SHIFTX2	www.shiftx2.ca
STING Millennium	sms.cbi.cnpia.embrapa.br/SMS/STINGm
SuperPose	wishart.biology.ualberta.ca/SuperPose
SWISS-MODEL	swissmodel.expasy.org
TargetDB	sbkb.org
TM-align	cssb.biology.gatech.edu/skolnick/webservice/TM-align/index.shtml
TopMatch	topmatch.services.came.sbg.ac.at
TopSearch	topsearch.services.came.sbg.ac.at
VADAR	vadar.wishartlab.com
VAST+	www.ncbi.nlm.nih.gov/Structure/vastplus/vastplus.cgi
WebMol	bioinformatics.mpimp-golm.mpg.de/group-members/mpi-mp-group/dirk-walther/webmol-1
WHAT_CHECK	swift.cmbi.umcn.nl/gv/whatcheck/

Further Reading

- Branden, C. and Tooze, J. (1999). *Introduction to Protein Structure*, 2e. New York, NY: Garland Science Publishing. A superb, easy-to-read reference with excellent coverage and great color diagrams. This book covers the field very nicely and, even though it was published nearly 20 years ago, just about every practicing structural biologist has a copy of either the first or second edition.
- Kelley, L.A. and Sternberg, M.J.E. (2009). Protein structure prediction on the web: a case study using the Phyre server. *Nat. Protoc.* 4: 363–371. A very detailed and helpful description of how to use the Phyre structure prediction server and how it works. This article also provides some excellent background material on protein structure prediction and a nice, balanced assessment of the strengths and weaknesses of structure prediction.
- Lesk, A.M. (2000). *Introduction to Protein Architecture: The Structural Biology of Proteins*. Oxford, UK: Oxford University Press. Another excellent book by Dr. Lesk. Beautifully illustrated and

very accessible to readers of all backgrounds. Provides many interesting problems and web-based exercises.

Rhodes, G. (2006). *Crystallography Made Crystal Clear: A Guide for Users of Macromolecular Models*, 3e. Cambridge, MA: Academic Press. A great introduction to protein X-ray crystallography for non-crystallographers. Explains many complex concepts in a clear, understandable manner. Also provides a very readable set of chapters on analyzing NMR structures, working with homology models, and visualizing protein structures.

References

- Bai, X.C., McMullan, G., and Scheres, S.H. (2015). How cryo-EM is revolutionizing structural biology. *Trends Biochem. Sci.* 40: 49–57.
- Bates, P.A., Kelley, L.A., MacCallum, R.M., and Sternberg, M.J. (2001). Enhancement of protein modeling by human intervention in applying the automatic programs 3D-JIGSAW and 3D-PSSM. *Proteins* (Suppl 5): 39–46.
- Bernstein, F.C., Koetzle, T.F., Williams, G.J.B. et al. (1977). The Protein Data Bank. *J. Mol. Biol.* 112: 535–542.
- Bonneau, R., Tsai, J., Ruczinski, I. et al. (2001). Rosetta in CASP4: progress in ab initio protein structure prediction. *Proteins* (Suppl 5): 119–126.
- Borrell, B. (2009). Fraud rocks protein community. *Nature* 462: 970.
- Bowie, J.U., Luthy, R., and Eisenberg, D. (1991). A method to identify protein sequences that fold into a known 3-dimensional structure. *Science* 253: 164–170.
- Bryant, S.H. and Lawrence, C.E. (1993). An empirical energy function for threading a protein sequence through a folding motif. *Proteins* 16 (1): 92–112.
- Brylinski, M. and Lingam, D. (2012). eThread: a highly optimized machine learning-based approach to meta-threading and the modeling of protein tertiary structures. *PLoS One* 7: e50200.
- Cavanagh, J., Fairbrother, W.J., Palmer, A.G. III, et al. (2006). *Protein NMR Spectroscopy: Principles and Practice*, 2e. Cambridge, MA: Academic Press.
- Chandonia, J.M., Fox, N.K., and Brenner, S.E. (2017). SCOPe: manual curation and artifact removal in the structural classification of proteins – extended database. *J. Mol. Biol.* 429: 348–355.
- Chou, P.Y. and Fasman, G.D. (1974). Prediction of protein conformation. *Biochemistry* 13: 222–245.
- Corey, R.B. and Pauling, L. (1953). Molecular models of amino acids, peptides, and proteins. *Rev. Sci. Instrum.* 24: 621–627.
- Davis, I.W., Leaver-Fay, A., Chen, V.B. et al. (2007). MolProbity: all-atom contacts and structure validation for proteins and nucleic acids. *Nucleic Acids Res.* 35 (Web Server issue): W375–W383.
- Dietmann, S., Park, J., Notredame, C. et al. (2001). A fully automatic evolutionary classification of protein folds: Dali domain dictionary version 3. *Nucleic Acids Res.* 29: 55–57.
- Doreleijers, J.F., Sousa da Silva, A.W., Krieger, E. et al. (2012). CING: an integrated residue-based structure validation program suite. *J. Biomol. NMR* 54: 267–283.
- Drenth, J. (2006). *Principles of Protein X-Ray Crystallography*, 3e. New York, NY: Springer.
- Gibson, K.D. and Scheraga, H.A. (1967). Minimization of polypeptide energy I. Preliminary structures of bovine pancreatic ribonuclease s-peptide. *Proc. Natl. Acad. Sci. U.S.A.* 58: 420–427.
- Hagen, J.B. (2000). The origins of bioinformatics. *Nat. Rev. Genet.* 1: 231–236.
- Hall, S.R., Allen, A.H., and Brown, I.D. (1991). The crystallographic information file (CIF): a new standard archive file for crystallography. *Acta Crystallogr. Sec. A: Found. Crystallogr.* 47: 655–685.
- Han, B., Liu, Y., Ginzinger, S.W., and Wishart, D.S. (2011). SHIFTX2: significantly improved protein chemical shift prediction. *J. Biomol. NMR* 50: 43–57.
- Hanson, R.M., Prilusky, J., Renjian, Z. et al. (2013). JSmol and the next-generation web-based representation of 3D molecular structure as applied to Proteopedia. *Isr. J. Chem.* 53: 207–216.

- Herráez, A. (2006). Biomolecules in the computer: Jmol to the rescue. *Biochem. Mol. Biol. Educ.* 34: 255–261.
- Higa, R.H., Togawa, R.C., Montagner, A.J. et al. (2004). STING Millennium suite: integrated software for extensive analyses of 3d structures of proteins and their complexes. *BMC Bioinf.* 5: 107.
- Hodis, E., Prilusky, J., Martz, E. et al. (2008). Proteopedia – a scientific “wiki” bridging the rift between three-dimensional structure and function of biomacromolecules. *Genome Biol.* 9: R121.
- Hooft, R.W., Vriend, G., Sander, C., and Abola, E.E. (1996). Errors in protein structures. *Nature* 381: 272.
- Kabsch, W. and Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22: 2577–2637.
- Källberg, M., Margaryan, G., Wang, S. et al. (2014). RaptorX server: a resource for template-based protein structure modeling. *Methods Mol. Biol.* 1137: 17–27.
- Kaplan, W. and Littlejohn, T.G. (2001). Swiss-PDB viewer (Deep View). *Briefings Bioinf.* 2: 195–197.
- Kelley, L.A., Mezulis, S., Yates, C.M. et al. (2015). The Phyre2 web portal for protein modeling, prediction and analysis. *Nat. Protoc.* 10: 845–858.
- Kendrew, J.C., Bodo, G., Dintzis, H.M. et al. (1958). A three dimensional model of the myoglobin molecule obtained by x-ray analysis. *Nature* 181: 662–666.
- Kim, D.E., Chivian, D., and Baker, D. (2004). Protein structure prediction and analysis using the Robetta server. *Nucleic Acids Res.* 32 (Web Server issue): W526–W531.
- Klepeis, J.L., Lindorff-Larsen, K., Dror, R.O., and Shaw, D.E. (2009). Long-timescale molecular dynamics simulations of protein structure and function. *Curr. Opin. Struct. Biol.* 19: 120–127.
- Krissinel, E. and Henrick, K. (2004). Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallogr. Sect. D: Biol. Crystallogr.* 60: 2256–2268.
- Kuntal, B.K., Aparoy, P., and Reddanna, P. (2010). EasyModeller: a graphical interface to MODELLER. *BMC Res. Notes* 3: 226.
- Laskowski, R.A., MacArthur, M.W., Moss, D.S., and Thornton, J.M. (1993). PROCHECK: a program to check the stereochemical quality of protein structures. *J. Appl. Crystallogr.* 26: 283–291.
- Levitt, M. (2007). Growth of novel protein structural data. *Proc. Natl. Acad. Sci. U.S.A.* 104: 3183–3188.
- Levitt, M. and Chothia, C. (1976). Structural patterns in globular proteins. *Nature* 261: 552–558.
- Lindorff-Larsen, K., Piana, S., Dror, R.O., and Shaw, D.E. (2011). How fast-folding proteins fold. *Science* 334: 517–520.
- Liu, Y. and Kuhlman, B. (2006). RosettaDesign server for protein design. *Nucleic Acids Res.* 34 (Web Server issue): W235–W238.
- Lüthy, R., Bowie, J.U., and Eisenberg, D. (1992). Assessment of protein models with three-dimensional profiles. *Nature* 356: 83–85.
- Lyskov, S., Chou, F.C., Conchúir, S.Ó. et al. (2013). Serverification of molecular modeling applications: the Rosetta online server that includes everyone (ROSIE). *PLoS One* 8: e63906.
- Madej, T., Boguski, M.S., and Bryant, S.H. (1995). Threading analysis suggests that the obese gene product may be a helical cytokine. *FEBS Lett.* 373: 13–18.
- Madej, T., Lanczycki, C.J., Zhang, D. et al. (2014). MMDB and VAST+: tracking structural similarities between macromolecular complexes. *Nucleic Acids Res.* 42 (Database issue): D297–D303.
- Maiti, R., Van Domselaar, G.H., Zhang, H., and Wishart, D.S. (2004). SuperPose: a simple server for sophisticated structural superposition. *Nucleic Acids Res.* 32 (Web Server issue): W590–W594.
- Marks, D.S., Colwell, L.J., Sheridan, R. et al. (2011). Protein 3D structure computed from evolutionary sequence variation. *PLoS One* 6 (12): e28766.
- Marti-Renom, M.A., Stuart, A.C., Fiser, A. et al. (2000). Comparative protein structure modeling of genes and genomes. *Annu. Rev. Biophys. Biomol. Struct.* 29: 291–325.

- Martz, E. (2002). Protein explorer: easy yet powerful macromolecular visualization. *Trends Biochem. Sci.* 27: 107–109.
- McCree, D.E. (1999). *Practical Protein Crystallography*, 2e. Cambridge, MA: Academic Press.
- Montomerie, S., Cruz, J.A., Shrivastava, S. et al. (2008). PROTEUS2: a web server for comprehensive protein structure prediction and structure-based annotation. *Nucleic Acids Res.* 36 (Web Server issue): W202–W209.
- Murzin, A.G., Brenner, S.E., Hubbard, T., and Chothia, C. (1995). SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.* 247: 536–540.
- NCBI Resource Coordinators (2017). Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.* 45 (D1): D12–D17.
- Nielsen, M., Lundegaard, C., Lund, O., and Petersen, T.N. (2010). CPHmodels-3.0 – remote homology modeling using structure-guided sequence profiles. *Nucleic Acids Res.* 38 (Web Server issue): W576–W581.
- Pearl, F.M.G., Lee, D., Bray, J.E. et al. (2000). Assigning genomic sequences to CATH. *Nucleic Acids Res.* 28: 277–282.
- Pieper, U., Webb, B.M., Dong, G.Q. et al. (2014). ModBase, a database of annotated comparative protein structure models and associated resources. *Nucleic Acids Res.* 42 (Database issue): D336–D346.
- Prlc, A., Bliven, S., Rose, P.W. et al. (2010). Pre-calculated protein structure alignments at the RCSB PDB website. *Bioinformatics* 26: 2983–2985.
- Ramachandran, G.N., Ramakrishnan, C., and Sasisekharan, V. (1963). Stereochemistry of polypeptide chain configurations. *J. Mol. Biol.* 7: 95–99.
- Read, R.J., Adams, P.D., Arendall, W.B. 3rd, et al. (2011). A new generation of crystallographic validation tools for the protein data bank. *Structure* 19: 1395–1412.
- Richards, F.M. (1977). Areas, volumes, packing and protein structure. *Annu. Rev. Biophys. Bioeng.* 6: 151–176.
- Richardson, J.S. (1981). The anatomy and taxonomy of protein structure. *Adv. Protein Chem.* 34: 167–339.
- Rose, A.S. and Hildebrand, P.W. (2015). NGL viewer: a web application for molecular visualization. *Nucleic Acids Res.* 43 (Web Server issue): W576–W579.
- Sali, A. (1998). 100,000 protein structures for the biologist. *Nat. Struct. Biol.* 5: 1029–1032.
- Sayle, R.A. and Milner-White, E.J. (1995). RASMOL: biomolecular graphics for all. *Trends Biochem. Sci.* 20: 374–376.
- Schaeffer, R.D. and Daggett, V. (2011). Protein folds and protein folding. *Protein Eng. Des. Sel.* 24: 11–19.
- Schwede, T., Kopp, J., Guex, N., and Peitsch, M.C. (2003). SWISS-MODEL: an automated protein homology-modeling server. *Nucleic Acids Res.* 31: 3381–3385.
- Sheffler, W. and Baker, D. (2010). RosettaHoles2: a volumetric packing measure for protein structure refinement and validation. *Protein Sci.* 19: 1991–1995.
- Shindyalov, I.N. and Bourne, P.E. (2001). A database and tools for 3-D protein structure comparison and alignment using the combinatorial extension (CE) algorithm. *Nucleic Acids Res.* 29: 228–229.
- Sippl, M.J. and Wiederstein, M. (2008). A note on difficult structure alignment problems. *Bioinformatics* 24: 426–427.
- Söding, J., Biegert, A., and Lupas, A.N. (2005). The HHpred interactive server for protein homology detection and structure prediction. *Nucleic Acids Res.* 33 (Web Server issue): W244–W248.
- Vaguine, A.A., Richelle, J., and Wodak, S.J. (1999). SFCHECK: a unified set of procedures for evaluating the quality of macromolecular structure-factor data and their agreement with the atomic model. *Acta Crystallogr. Sect. D: Biol. Crystallogr.* 55: 191–205.
- Vallat, B.K., Pillardy, J., Májek, P. et al. (2009). Building and assessing atomic models of proteins from structural templates: learning and benchmarks. *Proteins* 76: 930–945.

- Varadi, M., Kosol, S., Lebrun, P. et al. (2014). pE-DB: a database of structural ensembles of intrinsically disordered and of unfolded proteins. *Nucleic Acids Res.* 42 (Database issue): D326–D335.
- Walther, D. (1997). WebMol – a Java based PDB viewer. *Trends Biochem. Sci.* 22: 274–275.
- Wang, B., Wang, Y., and Wishart, D.S. (2010). A probabilistic approach for validating protein NMR chemical shift assignments. *J. Biomol. NMR* 47: 85–99.
- Westbrook, J.D., Feng, Z., Chen, L. et al. (2003). The Protein Data Bank and structural genomics. *Nucleic Acids Res.* 31: 489–491.
- Westbrook, J.D., Ito, N., Nakamura, H. et al. (2005). PDBML: the representation of archival macromolecular structure data in XML. *Bioinformatics* 21: 988–992.
- Wiederstein, M., Gruber, M., Frank, K. et al. (2014). Structure-based characterization of multiprotein complexes. *Structure* 22: 1063–1070.
- Willard, L., Ranjan, A., Zhang, H. et al. (2003). VADAR: a web server for quantitative evaluation of protein structure quality. *Nucleic Acids Res.* 31: 3316–3319.
- Wu, S. and Zhang, Y. (2007). LOMETS: a local meta-threading-server for protein structure prediction. *Nucleic Acids Res.* 35: 3375–3382.
- Wu, S. and Zhang, Y. (2008). MUSTER: improving protein sequence profile-profile alignments by using multiple sources of structure information. *Proteins* 72: 547–556.
- Yang, J. and Zhang, Y. (2015). I-TASSER server: new development for protein structure and function predictions. *Nucleic Acids Res.* 43 (Web Server issue): W174–W181.
- Ye, Y. and Godzik, A. (2004). FATCAT: a web server for flexible structure comparison and structure similarity searching. *Nucleic Acids Res.* 32 (Web Server issue): W582–W585.
- Young, J.Y., Westbrook, J.D., Feng, Z. et al. (2017). OneDep: unified wwPDB system for deposition, biocuration, and validation of macromolecular structures in the PDB archive. *Structure* 25: 536–545.
- Zhang, Y. and Skolnick, J. (2005). TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.* 33 (7): 2302–2309.

13

Biological Networks and Pathways

Gary D. Bader

Introduction

A major challenge for biologists is to gain an understanding of the workings of the cell by integrating available information from the various fields of molecular and cellular biology into an accurate cellular model that can be used to generate hypotheses for testing. In particular, the exponentially growing amount of data from now routine comprehensive measurements of DNA, RNA, and proteins in biological samples provides rich opportunities to discover novel biological functions, genotype–phenotype correlations, and the underlying mechanisms of disease causation. Excitingly, experimental methods, such as transcript expression-level measurement using RNA-seq methods (Chapter 10) and protein identification by mass spectrometry (Chapter 11), are increasingly sensitive and can detect tens of thousands of molecules within a biological sample at decreasing cost. This has led to the collection of vast amounts of data about biological systems. However, the analysis and interpretation of all these data is a major challenge for many researchers. Analyses often highlight large lists of genes that may require an impractically large amount of manual literature searching to interpret. Biological pathway and network analysis provides a useful approach to address this data integration, modeling, and interpretation challenge. Pathway and network analysis methods use information about pathways (representing detailed biological processes) and also networks (generally representing molecular interaction networks, such as protein–protein or protein–DNA interaction networks) to aid data interpretation.

To illustrate the useful insights into biological mechanisms that can be achieved using pathway and network analysis, consider two successful examples. Pathway analysis was used to identify histone and DNA methylation by the polycomb repressive complex (PRC2) as the first rational therapeutic target for ependymoma, prevalent among childhood brain cancers (Mack et al. 2014). This pathway is targetable by available drugs, such as 5-azacytidine, which was used on a compassionate basis in a terminally ill patient and stopped rapid metastatic tumor growth. This promising result led to the initiation of two clinical trials. In another example, pathway analysis of rare copy number variant data in autism identified several significant pathways affected by gene deletions, whereas only a few significant hits were identified with case–control association tests of single genes or genomic loci (Pinto et al. 2010). The inclusion of pathway information increased the statistical power of the analysis approach and uncovered otherwise hidden aspects of biology in both these disease areas.

As brief historical context, the pathway informatics field started in the 1990s with work on the computational representation of metabolic pathways (Karp and Riley 1993). Biological network informatics was introduced in the early 2000s, necessitated by the first large-scale cellular protein–protein and genetic interaction maps (Ito et al. 2000; Schwikowski et al. 2000; Walhout et al. 2000; Tong et al. 2001). Pathway enrichment analysis of large gene lists, now the most popular type of pathway analysis, was introduced in the late 1990s (Tavazoie et al. 1999) and popularized in the mid-2000s.

While well established, pathway and network data continue to grow, and the field is still being heavily researched. Given the changing nature of the field, this chapter covers a number of useful and freely available tools and methods, but also focuses on fundamental theory that should be applicable to new resources as they become available. General theory is covered first, then tools. The first topic covered is pathway and molecular interaction data, including where the data come from and how they are represented, stored, and accessed. The second topic is pathway and network visualization and analysis, covering fundamental concepts and the most popular and useful analysis methods and tools. The analysis methods selected are meant to illustrate the interesting biological questions that can be answered by integrating pathway and network data with other data types, but do not cover all areas of the field in depth. Thus, pointers to online descriptions and lists of other pathway- and network-related databases and software tools are provided where relevant throughout the chapter.

Pathway and Molecular Interaction Mapping: Experiments and Predictions

Before explaining how pathway and molecular interaction data are stored and used, it is important to know what types of data exist and how these data are collected. What would an ideal biological experiment tell us? The answer is no less than everything: which molecules are in the cell at what time and at what place, how many molecules there are, which molecules they interact with, and their interaction dynamics. Ideally, one would want this information not only over the course of the cell cycle and other time-based cellular processes, but also in all important environmental conditions and under all known disease states. A wide range of biochemical, molecular biological, and genetic experiments have been invented to help elucidate cellular systems and determine which cellular parts are involved and how they fit together. However, current experimental methods, while useful and growing better every year, only scratch the surface of what is actually happening inside a cell or tissue. They usually only cover one layer of information (e.g. protein–protein interactions) and are mostly incomplete (Pouliot and Karp 2007; Braun et al. 2009). This is important to consider while using pathway and network information.

Metabolic pathways are the oldest of the pathway models and are composed of a series of enzymatic reactions. Enzymatic reactions have been studied for centuries, initially examining processes such as fermentation. The basic principle of experimentally mapping metabolism, composed mainly of protein enzymes, is to identify an enzymatic process (e.g. the conversion of glucose to ethanol in yeast) and progressively purify cellular extracts to find the enzymes involved. Validation involves testing whether the purified enzyme can convert the given substrate to a product. This process requires protein separation and purification technology, as well as molecule identification methods to identify the enzyme, cofactors, substrates, and products involved in the reaction.

Major advances in this area have been made using various forms of chromatography, gel-based separation techniques, nuclear magnetic resonance (NMR), and mass spectrometry (see Chapters 11 and 14). Chromatography and gel separation work on the basic principle that a molecular mixture can be decomposed based on component physicochemical properties such as size or charge. NMR and mass spectrometry can be used to directly identify small molecules and proteins based on atomic distance measurements and mass, respectively. Enzymologists further characterize the reaction rates of enzymes (kinetics) and the detailed enzymatic mechanism involved in catalysis (Voet and Voet 2004).

Unlike metabolic pathways, signaling pathways, which involve a higher proportion of direct protein–protein relationships such as the phosphorylation of one protein by another (a protein kinase), can be mapped using protein–protein interaction detection methods. Other molecule types, such as lipids and small molecules, are also involved in signaling, so protein interactions themselves only tell part of the story. Many techniques for determining protein–protein

interactions have been developed over the past few decades. One popular class of experiments uses co-purification, based on the methods described above. Proteins that strongly interact will purify as a complex that can be further degraded using harsher purification conditions to finally separate and identify the complex components. Importantly, this means that the definition of a protein complex depends on the purification conditions used, which measure a continuum of protein–protein interaction strengths. An example of a modern biochemical co-purification uses affinity chromatography to purify a protein complex from a cellular extract, then identifies the resulting complex components using mass spectrometry (see Chapter 11). Yeast two-hybrid methods are often used to determine whether two proteins can interact. An activation and DNA binding domain of a transcription factor are attached to each protein, respectively. If the two proteins of interest interact, the activation and DNA binding domains will also interact, forming a functional transcription factor that will express an engineered reporter gene. Presence of the reporter gene indicates binding. Major projects are in progress to use this technology to comprehensively map interactions in human cells and tissues (Luck et al. 2017).

Another often-used method is molecular cross-linking, an experimental method in which a linear molecule of defined length having two reactive ends is added to a mixture containing a potential complex in order to cross-link proteins that are close together; the distance over which an interaction can be detected is determined by the length of the cross-linker being used (Li et al. 2017). Subsequent purification and definition of the complex is easier, as the protein complex is covalently tied together instead of just being electrostatically bound. Many other protein–protein interaction-determining experimental methods exist (Phizicky and Fields 1995). Each experiment has its own strengths and weaknesses, and multiple types of experiments must be performed to increase confidence that the result is relevant *in vivo*.

As these kinds of experiments are often expensive and time-consuming, many computational methods have been developed for predicting pathways and interactions. These are rarely as accurate as detailed “wet lab” experimental analysis, thus should be treated as hypotheses that require experimental testing. However, these methods can rapidly predict information, frequently with high levels of accuracy; this is especially useful in cases where experiments are infeasible, such as in organisms that cannot easily be studied experimentally.

Metabolic pathways can be accurately predicted by mapping the proteins (enzymes) in a known pathway from one organism to another organism using orthology relationships, with additional steps applied to construct pathways from the results. This is feasible because metabolic pathways are frequently highly conserved between species, thus reasonably accurate predictions can be made. Many rules can be used for predicting metabolic pathways. For instance, key reactions must be present. Enzymes that are part of multiple pathways cannot be considered to unambiguously indicate the presence of a pathway. Pathways are then validated by checking that they balance in terms of input and output mass. If they do not balance or are disconnected because of missing enzymes, these enzymes can be more thoroughly searched in the genome being annotated using a process termed hole filling. As is usual for orthology-based prediction methods, results for the reconstruction will be better when the experimentally known pathways being used are from a species that is close to the one being annotated. The PathoLogic algorithm (Karp et al. 2011) used by the BioCyc family of databases (see section EcoCyc) uses this method to predict entire pathway databases for an organism, based on the organism’s genome. The end result of this prediction system is an excellent draft metabolic model for an organism, though manual curation is required to fix errors in the resulting pathways to achieve a high-quality model. Signaling and gene regulation pathways cannot currently be accurately predicted in this manner because they are much less conserved than metabolic pathways.

Molecular interactions can be predicted using a range of methods. Predicted molecular interactions ideally represent direct physical binding, though most molecular interaction prediction methods do not guarantee the prediction of a direct physical interaction. For instance, predictions may include “functional interactions,” which are interactions between proteins

within the same pathway or among genes with similar function. This is still useful because the likelihood of protein interaction within a pathway or function is higher than that among randomly selected proteins. A range of patterns have been correlated with protein–protein interactions and these can be used to predict these interactions. Genes that have similar transcription profiles have been shown to physically interact more often than expected (Ge et al. 2001; Grigoriev 2001; Jansen et al. 2002) and this effect is stronger with protein expression profiles (Kim et al. 2014). Protein interactions can also be mapped across species. If an interaction between two proteins is known in one organism, it may be possible to successfully predict that their orthologs bind in another organism (Matthews et al. 2001; Tien et al. 2004), though this pattern is more relevant for conserved proteins and protein complexes (Brown and Jurisica 2007). Genes whose protein products physically interact are sometimes maintained in close physical chromosomal proximity to each other (Tamames et al. 1997; Dandekar et al. 1998; Overbeek et al. 1999). The most obvious case of this phenomenon are operons in bacteria and archaea, where genes whose protein products function in the same biological process are transcribed on the same polycistronic messenger RNA (mRNA). Two of the main driving forces in genome evolution are gene genesis and loss (Snel et al. 2002). The fact that a gene pair remains together across many different species often represents a concerted evolutionary effort to keep them together, as might be the case if they functioned in the same biological process. Phylogenetic profiles show the presence or absence of genes across complete genomes from many species (Ouzounis and Kyrpides 1996; Rivera et al. 1998; Pellegrini et al. 1999), and pairs of genes that have very similar phylogenetic profiles are candidates for physical interaction. A gene fusion event represents the physical fusion of two separate parent genes into a single multi-functional gene. This is the ultimate form of gene co-localization: interacting genes are not just kept in close proximity in the genome but are physically joined as a single entity. It has been suggested that the driving force behind these events is to lower the regulation load of multiple interacting gene products (Enright et al. 1999). Hence, gene fusion events provide a way to computationally detect functional and physical interactions between proteins (Enright et al. 1999; Marcotte et al. 1999).

Each of the above computational methods has strengths and weaknesses. Gene neighborhood and phylogenetic profile methods give better predictions as the number of completely sequenced genomes they use increases. The gene fusion method predicts well but is not generalizable, as the actual number of detected fusion events is typically small. These genome-based methods tend to work better with prokaryotic genomes. Gene co-expression analysis is weakly predictive. All methods improve reliability when more data (e.g. genomes and gene or protein expression profiles) are used. One way to address this is to combine data from all available prediction methods using machine learning approaches. Each evidence source is automatically weighted based on its ability to accurately predict known interactions. The first protein–protein interaction prediction method of this type used Bayesian network machine learning to predict protein interactions in the budding yeast (Jansen et al. 2003). A probability value for a protein interaction can be calculated given the available evidence for it across all sources. Protein interactions predicted in this way have been shown to be as reliable as high-throughput experimental techniques and cover a larger proportion of genes (Kotlyar et al. 2015). Recent methods have extended this approach with more evidence sources, including ones that enable protein–protein interaction binding sites to be predicted, at least for certain protein classes (Jain and Bader 2016).

Pathway and Molecular Interaction Databases: An Overview

Given the breadth and depth of pathway and molecular interaction network information available from small- and large-scale experiments and its utility for understanding how biological systems work, it is no surprise that a number of databases have been built to represent and store this information. In fact, there are over 700 pathway and molecular interaction-related

database resources listed in the Pathguide link directory (Bader et al. 2006). These range widely in form and content and include full-featured pathway databases, ones that focus on protein–protein or other molecular interactions, and organism- or disease-specific pathway databases. This section first covers general theory about how pathways and interaction networks are defined and represented in databases, then describes a selection of the largest, most generally useful and freely available database resources.

Representing Biological Pathways and Interaction Networks in a Computer

The cell is a large, complex, and dynamic connected network of molecules. Because of its complexity, it is useful to organize the cell into substructures and subsystems such as organelles, pathways, and complexes, so as to aid in understanding the overall structure. While organelles and complexes can be directly observed under a microscope, pathways cannot, so it is important to realize that pathways are human models and are parts of a larger, connected molecular interaction network. Pathways can be considered a series of molecular interactions and reactions, often forming a network, that carry out some defined process. Pathways are often defined based on recognized biochemical or information-processing phenomena. For instance, a series of metabolic reactions could start with the intake of a metabolite from the environment that is converted irreversibly to something else. For example, the glycolysis pathway breaks down glucose to generate energy (adenosine triphosphate). Also, signal propagation in a series of signaling pathway steps could be shown to follow a specific pathway, such as when a ligand binds to a cell surface receptor that results in signal propagation through a protein kinase cascade in the cytoplasm and then to the nucleus, where a transcriptional response is activated. Often in these cases, the start- and endpoints of a pathway are defined by observation of a readily detectable phenotype after stimulation or perturbation, such as observing gene expression after stimulating the cell with a peptide growth hormone.

Pathways can be classified into different types, and each of the main types generally has a different computational representation in the various existing pathway databases. The main types of pathway representations model metabolic, signal transduction (also called cell signaling), and gene regulation pathways. Metabolic pathways are generally defined by a series of chemical actions and the chemical results of those actions, for the purpose of changing one molecular species into another (e.g. glycolysis). Signal transduction pathways are usually defined by binding events (e.g. protein–protein interactions) that sometimes involve chemical actions (e.g. phosphorylation events) for the purpose of communicating information from one place in the cell to another. The epidermal growth factor receptor pathway is a common example of a signal transduction pathway that conveys information from an externally activated cell surface receptor to the nucleus in order to effect change in gene expression in response to an external signal. Finally, gene regulation networks involve transcription factors or other regulators activating or repressing expression of genes, including other transcription factors. Each of these pathway types is often described using a characteristic representation style that includes convenient shorthand notation for more complex biological processes. For example, gene regulation is abstracted as a single relationship in a gene regulation pathway (e.g. “NOTCH regulates HES1”), whereas it would be a large, multi-step process when represented as a metabolic pathway. This can lead to difficulties when trying to integrate pathways represented differently from diverse sources. Thus, it is important to understand how a database represents the information it stores to be able to query it and to understand its advantages and limitations.

When molecular interactions are part of pathways, they are represented as protein complex formation events. Because high-throughput experimental methods exist to map thousands of molecular interactions, many of which are not part of pathways, a separate convenient shorthand representation has been developed for these interactions. Molecular interactions can occur between any molecule types. The interactions are generally represented as a binary (pairwise) relationship, though sometimes interactions involve more than two participants. The type of the interaction is automatically defined based on the participant types. For

instance, an interaction between two proteins is a protein–protein interaction. Molecular interaction representation schemes include the type and definition of the participating molecules, as well as details of the experiment used to determine the interaction. This is another difference between pathways and interactions – pathways describe a model developed based on many experiments and often the experimental details are not described along with the pathway, whereas molecular interactions are often directly determined by individual experiments that provide important information about the quality level of the interaction data. Collections of molecular interactions are represented as networks (see Network Visualization). An interactome is defined as the set of all interactions in a cell or organism, by analogy to a “genome.”

Considerations for Pathway and Interaction Data Representation

Since there are many ways of representing pathways and interactions, it is useful to review some basic data representation principles to better understand why representing this information is so complex compared with, for example, biological sequences (see Chapter 1). A representation system (also called a data model or abstraction) is an invention that can be used to describe and organize a set of information. Many different representation schemes are often possible for the same type of information, and two different people given the task to invent an abstraction independently can easily create different systems, especially for complex and partially undefined biological information-like pathways. A single representation scheme must be agreed upon before it can be useful for data communication, although such a decision involves considering a number of trade-offs. An ideal representation system compactly and efficiently describes exactly the information useful to the users of the system, facilitating communication among people with the same extensive common knowledge such as scientists in a specific subfield who all understand the jargon and concepts of their field. Compactness can be achieved because common knowledge can be taken for granted and, thus, does not have to be explicitly represented each time information is communicated. This compactness can enormously reduce communication time and effort, making it very useful. Unfortunately, using a compact representation to communicate between people who do not share the same common knowledge does not work as well. These people will have trouble understanding each other unless common knowledge is explicitly represented. This frequently happens when people in different subfields in science communicate. Similarly, computer programs that are not programmed with extensive rules defining common knowledge can generally not properly “understand” very compact representation, requiring the explicit coding of additional information and logic to perform actions such as querying or visualizing the compact data.

A related trade-off is between simplicity and complexity of representation. The advantage of having a simple model that captures the basic properties of the data is that it is easily created, understood, and used, but it cannot represent all of the detail that may be known about a system. The complex model may be able to represent everything that is known but might be too unwieldy to be useful in some cases. Many aspects of biological systems that may be useful to represent can significantly add to the complexity of a representation scheme. Examples are level of detail, context, and tracking where the original information comes from (its *provenance*, discussed further below). Each of these is dealt with individually in the next paragraph.

Adding levels of detail in data modeling is useful for representing data at varying levels of knowledge or understanding. When relevant details are known, a detailed data model should be able to represent them. In a model that includes multiple levels of detail, there is a choice between representing the same information at low, intermediate, or high detail levels. Depending on the goal, more or less detail may be required. For instance, we may know that a protein phosphorylation event is catalyzed by a tyrosine kinase at a specific amino acid position. Alternatively, someone studying the global properties of protein interaction networks may only be interested in the fact that one protein interacts with another and would find information on post-translational modifications distracting. Adding to the complexity of biological knowledge

representation, levels of detail in the cell map can be considered across large ranges (scales) of time and space, meaning that each level of the organizational hierarchy may require its own abstraction system. As an example across spatial scales, the molecular parts of the cell have widely established representation systems, such as the 20-letter amino acid code for protein sequence and the atoms, bond lengths (measured in angstroms, 10^{-10} m), and connectivity of atoms in a three-dimensional protein structure. Neither of these abstractions works well in describing larger substructures of the cell, such as the nucleus or whole cells (measured in micrometers and up to the meter-length scale for human neurons). Similarly, across temporal scales, ultra-fast electron flow in a biochemical reaction, measured in attoseconds (10^{-18} s), can be described when it is known, but any useful abstraction to describe electron flow would not be useful for describing events on the time scale of the cell cycle, measured in minutes to hours.

Contextual information is important because molecular interactions and reactions depend on the presence of the participating molecules at permissive conditions, such as being present in the same place at the same time in a cell. A reaction may or may not occur with the same participants in different cells, in different developmental stages, or in different organisms. Similarly, it may be useful to capture the experimental evidence for the pathway knowledge being represented, as well as where information used to define the pathway came from. This knowledge-tracking information is referred to as *provenance*, which simply means proof of origin and authenticity. Describing context, evidence, and provenance adds complexity to the representation model.

Pathway Databases

Reactome

Reactome is a curated database of human pathways (Fabregat et al. 2018). Reactome represents pathways using a biochemical paradigm that models pathways as collections of events of different types. Reactome is one of the largest human pathway databases and covers signaling, metabolism, gene regulation, and disease pathways; it also includes pathway information for over half the human proteome. Manually laid out graphical displays of each pathway are available (Figure 13.1). Data can be downloaded in a variety of formats and various pathway analysis and query systems are freely available.

EcoCyc

EcoCyc is a literature-derived and curated encyclopedia of *Escherichia coli* bacteria metabolism (strain K12; Keseler et al. 2017). It has the most comprehensive coverage of any species-specific metabolic pathway database (Figure 13.2). MetaCyc (Caspi et al. 2018) is another literature-derived, curated database covering a broad range of organisms; it contains information about pathways in thousands of species, including microorganisms, plants, and animals, with *E. coli* having the largest representation. BioCyc is a collection of pathway databases containing EcoCyc and MetaCyc, as well as additional metabolic pathway predictions for thousands of organisms with sequenced genomes (including human) made using the PathoLogic algorithm (Karp et al. 2011), as described in Pathway and Molecular Interaction Mapping: Experiments and Predictions. EcoCyc and MetaCyc are freely available, while access to the rest of the BioCyc databases requires a subscription, sometimes available via a university library. The Pathway Tools software, which can be downloaded freely by academics, is useful for creating a metabolic pathway database for a newly sequenced genome. Some databases have used Pathway Tools to curate their own organism-specific pathway databases (Evsikov et al. 2009).

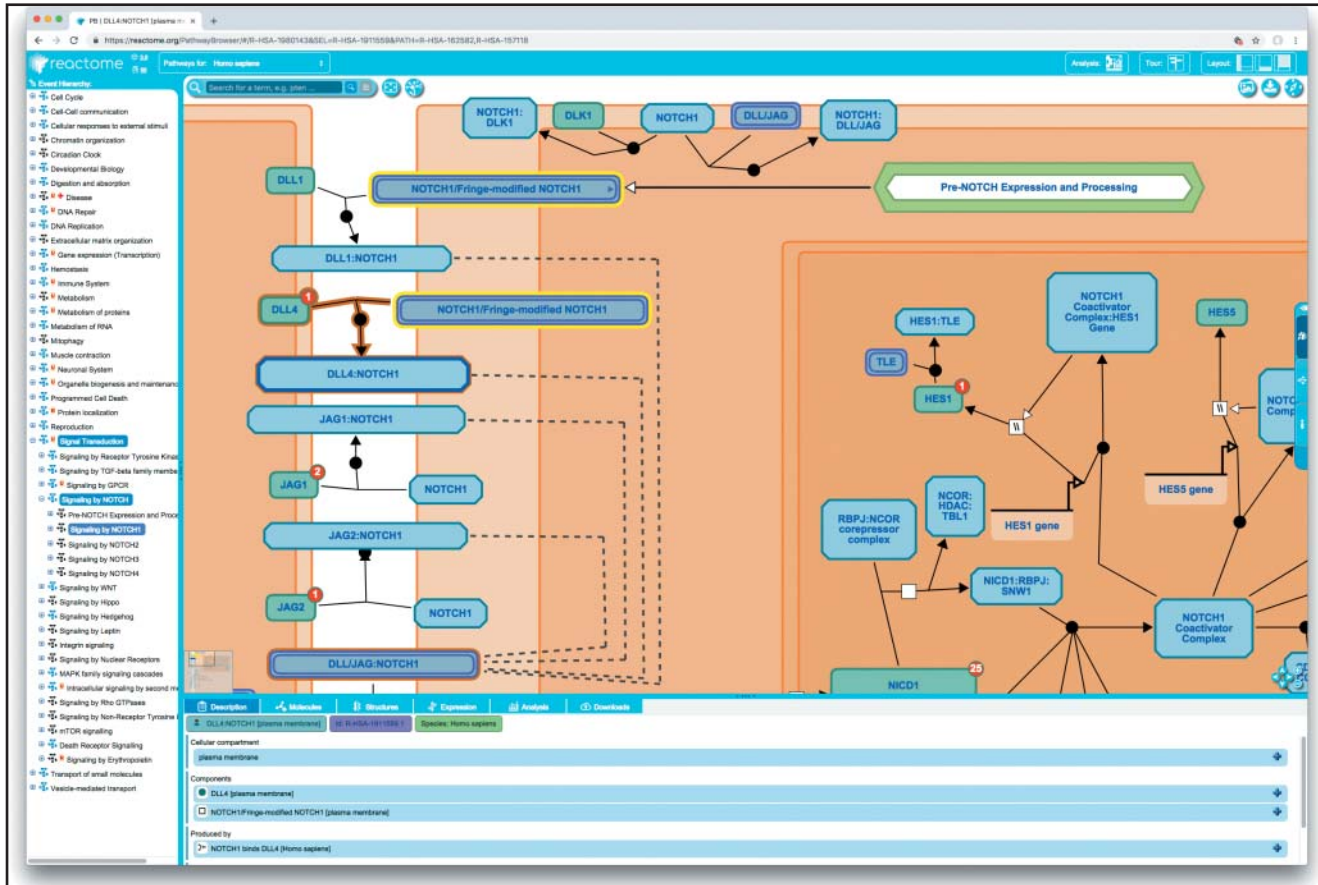


Figure 13.1 The Reactome database pathway view. The central view shows pathway participants such as proteins, genes, and complexes, each depicted as boxes. The reactions they participate in are depicted by various types of connection lines. The bottom panel shows information about participants in the main view that can be selected by clicking. Selected participants are highlighted in yellow and reactions are highlighted in brown. The numbers in the red circles on some participant boxes in the main window indicate that physical interactions involving those participants are available but are not shown. Clicking on the red circle toggles these interactions on and off. The left panel shows the hierarchical organizational view of all the pathways found within Reactome.

KEGG

The Kyoto Encyclopedia of Genes and Genomes (KEGG) pathway database contains curated metabolic, signaling, and disease pathways (Kanehisa et al. 2002). Information on enzymatic reactions, enzymes, small molecules, and genes is also available. Pathways are available as searchable and clickable images called maps, which KEGG is most known for (Figure 13.3a). Pathway maps can depict metabolism, regulatory pathways, and large complexes such as the ribosome, disease-related gene sets, and other gene collections. Each type of map has its own graphical representation style. Most metabolic pathway maps are reference maps that depict generalized pathways. Generalized pathways are not species specific, thus they might never be found in their entirety in a single species. Species-specific maps are automatically created by mapping reference pathways to a given species by orthology (Figure 13.3b). KEGG pathway maps link to a variety of underlying KEGG databases, including the LIGAND database for enzymes, reactions, and compounds, as well as to genome information. Pathways can be searched and browsed via the KEGG web site. The various interlinked KEGG databases are available freely via the World Wide Web, but download requires a license.

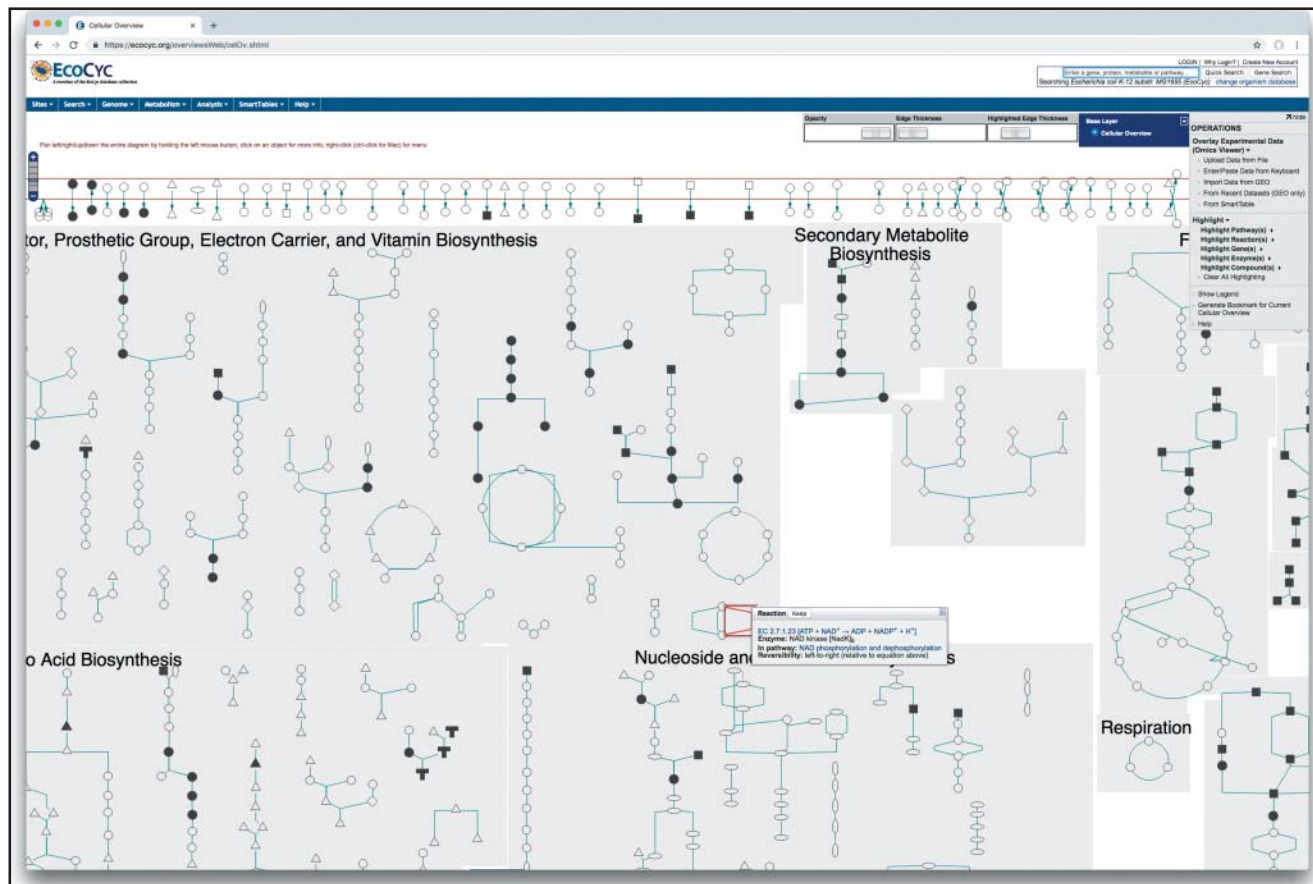


Figure 13.2 The EcoCyc database cellular overview of *Escherichia coli* metabolism. The central view shows a zoomed-in portion of the overview. Nodes represent metabolite molecules and lines represent reactions. Selecting molecules or reactions highlights them in red and produces a pop-up box containing further description. The top and right toolbars and menus provide links to a range of functionality available within EcoCyc and related databases (such as BioCyc).

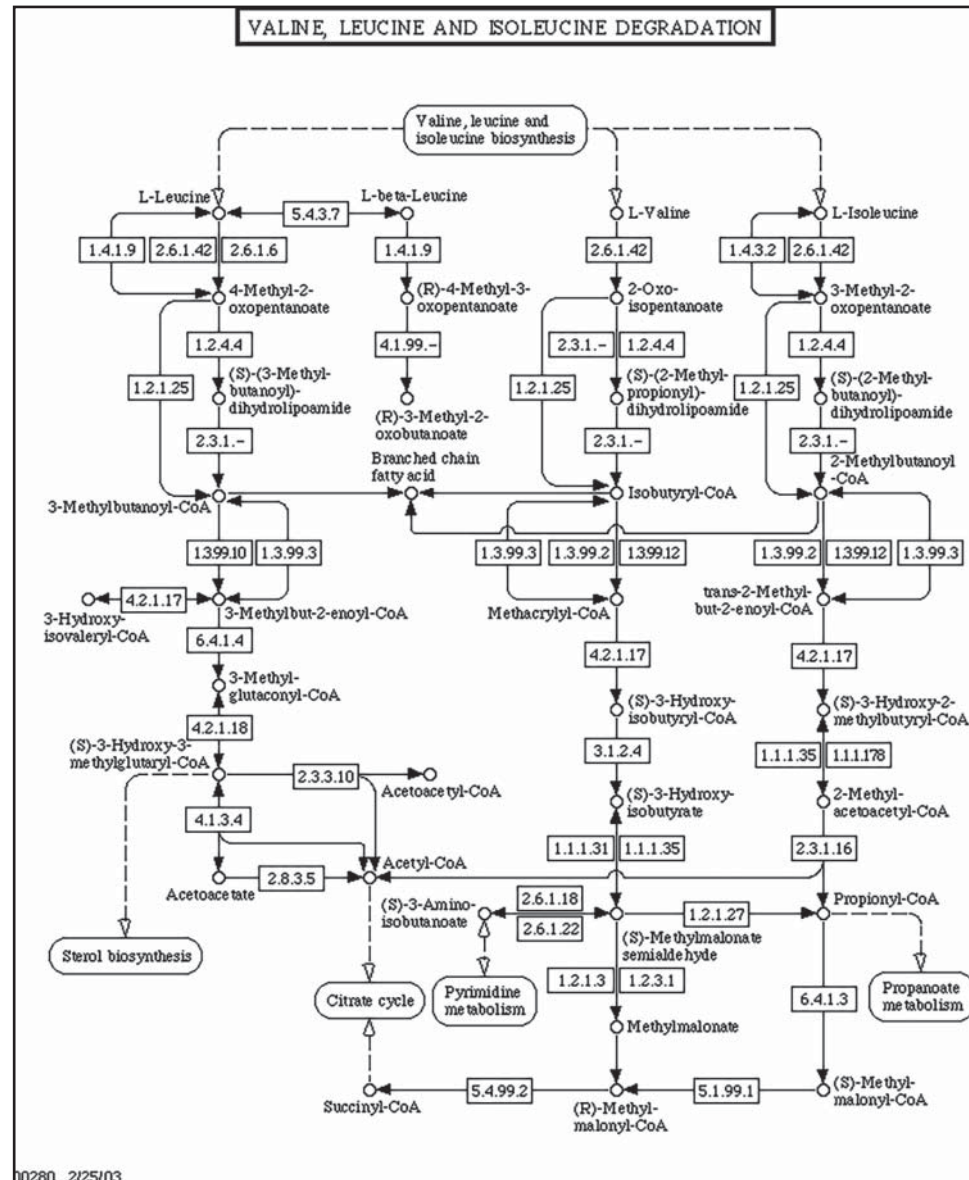
Molecular Interaction Databases

BioGRID

The Biological General Repository for Interaction Datasets, or BioGRID, contains manually curated protein–protein and genetic interactions, as well as chemical associations for a range of species (Chatr-Aryamontri et al. 2017). BioGRID is gene centric, which means the website organizes interactions around a single gene (Figure 13.4). Users can search for a gene of interest; for each gene, physical (i.e. protein–protein), genetic (e.g. synthetic lethal), and chemical (e.g. inhibition) interactions are presented in a table, along with experimental evidence for each interaction. Protein post-translational modification site and basic gene description information is also available. Data are freely available in Proteomics Standards Initiative–Molecular Interactions (PSI-MI) XML and tab-delimited text formats (see Standard Data Formats for Pathways and Molecular Interactions).

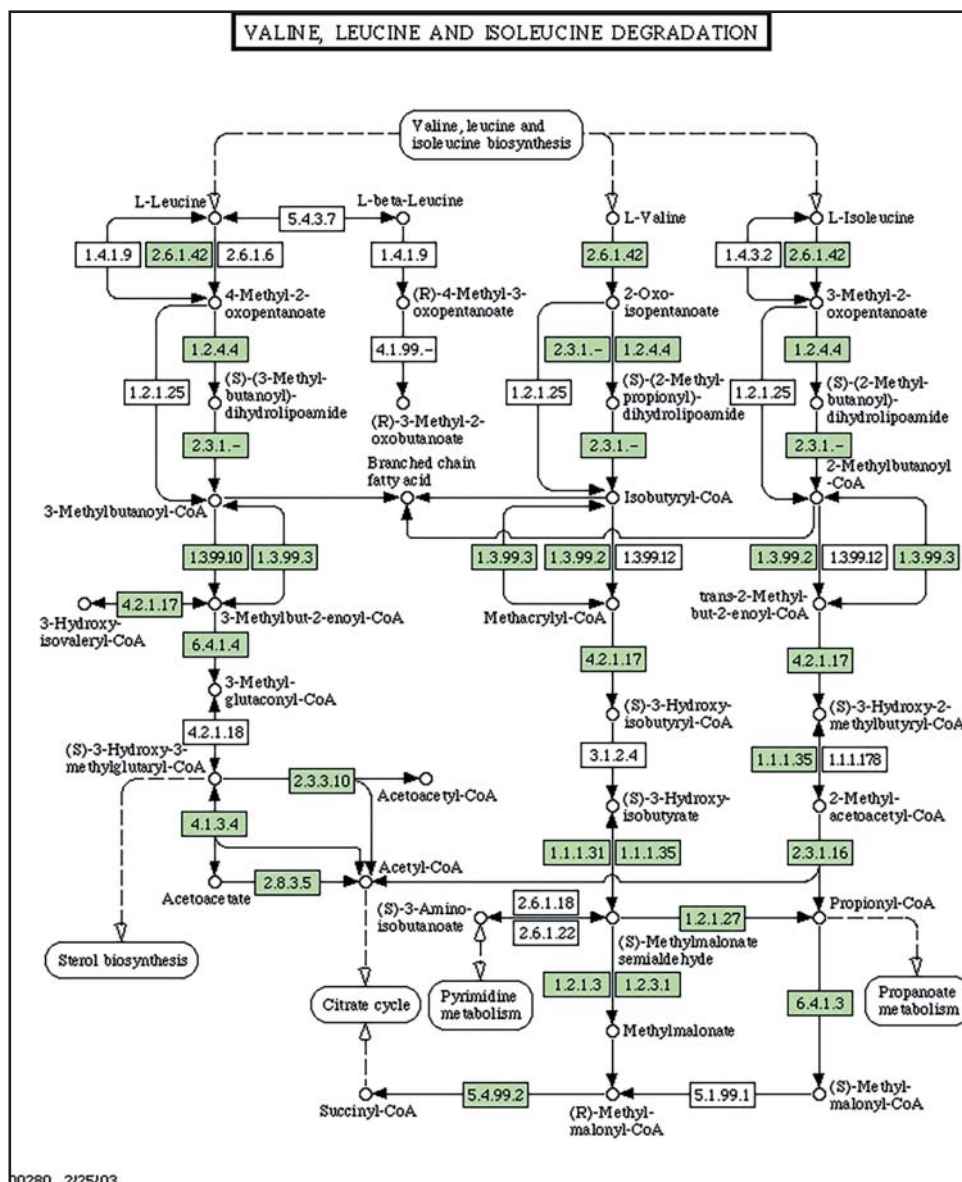
IntAct

IntAct, maintained by the European Bioinformatics Institute, is a protein interaction database that contains manually curated and user-submitted data. Database records are organized



(a)

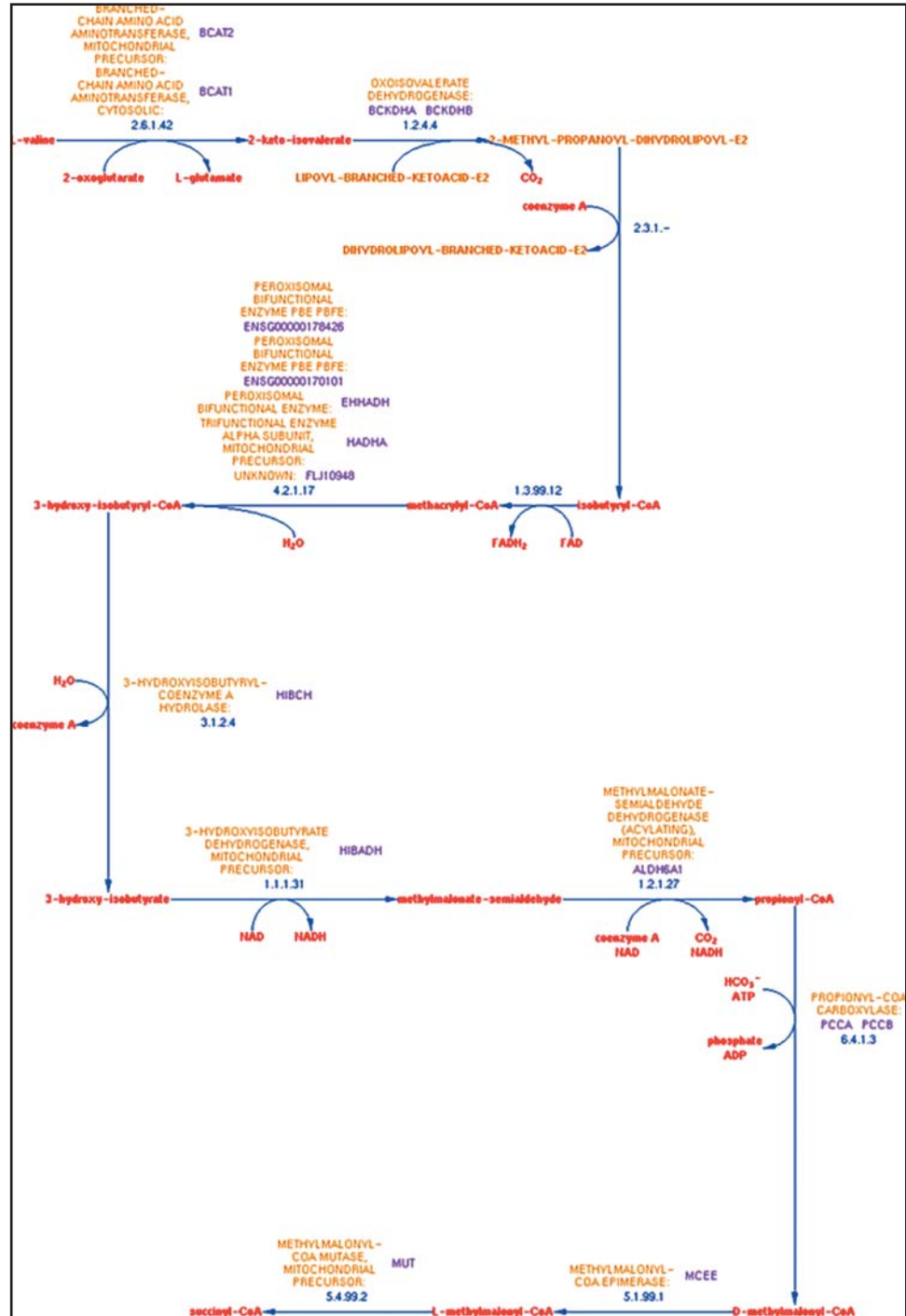
Figure 13.3 An example of metabolic pathway reconstruction from Kyoto Encyclopedia of Genes and Genomes (KEGG) and BioCyc: the valine degradation pathway reconstructed in human. (a) The reference valine degradation pathway in KEGG. The KEGG reference pathway is a superset of all known valine degradation pathway components from all organisms. (b) The enzymes that KEGG has found to be present in the sequenced human genome are highlighted in green. In KEGG, enzymes are represented by their Enzyme Commission (EC) number (e.g. 2.6.1.42), defining the enzyme's function. The EC system is a hierarchy of enzyme functions similar to the newer Gene Ontology molecular function-controlled vocabulary. Notice that not all enzymes from the reference pathway are highlighted in green. This is because KEGG was not able to find these enzymes in the human genome. A good example of this is the 3-hydroxyisobutyryl coenzyme A hydrolase (EC 3.1.2.4) that should exist in the human valine degradation pathway because there are no other enzymes from the reference pathway that can replace its function. Thus, this missing enzyme represents a "hole" in the pathway. This does not mean that the enzyme does not exist in the human genome. It may not be easily recognized because of sequence divergence over evolutionary time or because of inaccurate gene finding. The HumanCyc pathway reconstruction from the BioCyc family of databases is able to fill the hole (c). Notice that the EC 3.1.2.4 enzyme is present and linked to the *HIBCH* gene. Clicking on this gene in HumanCyc produces links to various sequence databases that contain this gene, as well as to publications that provide evidence that the *HIBCH* gene is an EC 3.1.2.4 enzyme. The extra computational and curatorial effort by HumanCyc resulted in holes being filled.



(b)

Figure 13.3 (Continued)

around interactions, experiments, and publications, and a graphical network viewer is available (Figure 13.5). One difference in the IntAct data model as compared with most other protein–protein interaction databases is that interactions can have more than two participants. The advantage of using sets to store interactions is that they can represent proteomics-derived protein complex data where the set of proteins that co-purifies is known but the direct physical interactions among these entities may be unknown (Gavin et al. 2002; Ho et al. 2002). The disadvantage of using sets is that interaction data represented as a set of more than two participants must be mapped to pairwise interactions for network visualization and analysis. This is frequently performed using a “spoke” expansion, where the experimental bait protein is linked to all proteins identified in the purification experiment, even if they may not directly physically interact (Bader and Hogue 2002). IntAct also maintains a database of curated protein complexes (Meldal et al. 2015). Data are freely available in PSI-MI XML and tab-delimited text formats.



(c)

Figure 13.3 (Continued)

Functional Interaction Databases

Functional interactions link genes if they are expected to have similar functions, where “function” can be defined in many ways. Functional interaction databases collect or predict large amounts (millions) of these links from a variety of sources. These resources are useful for exploring the function of a gene (or set of genes) by examining the function of other genes it interacts with.

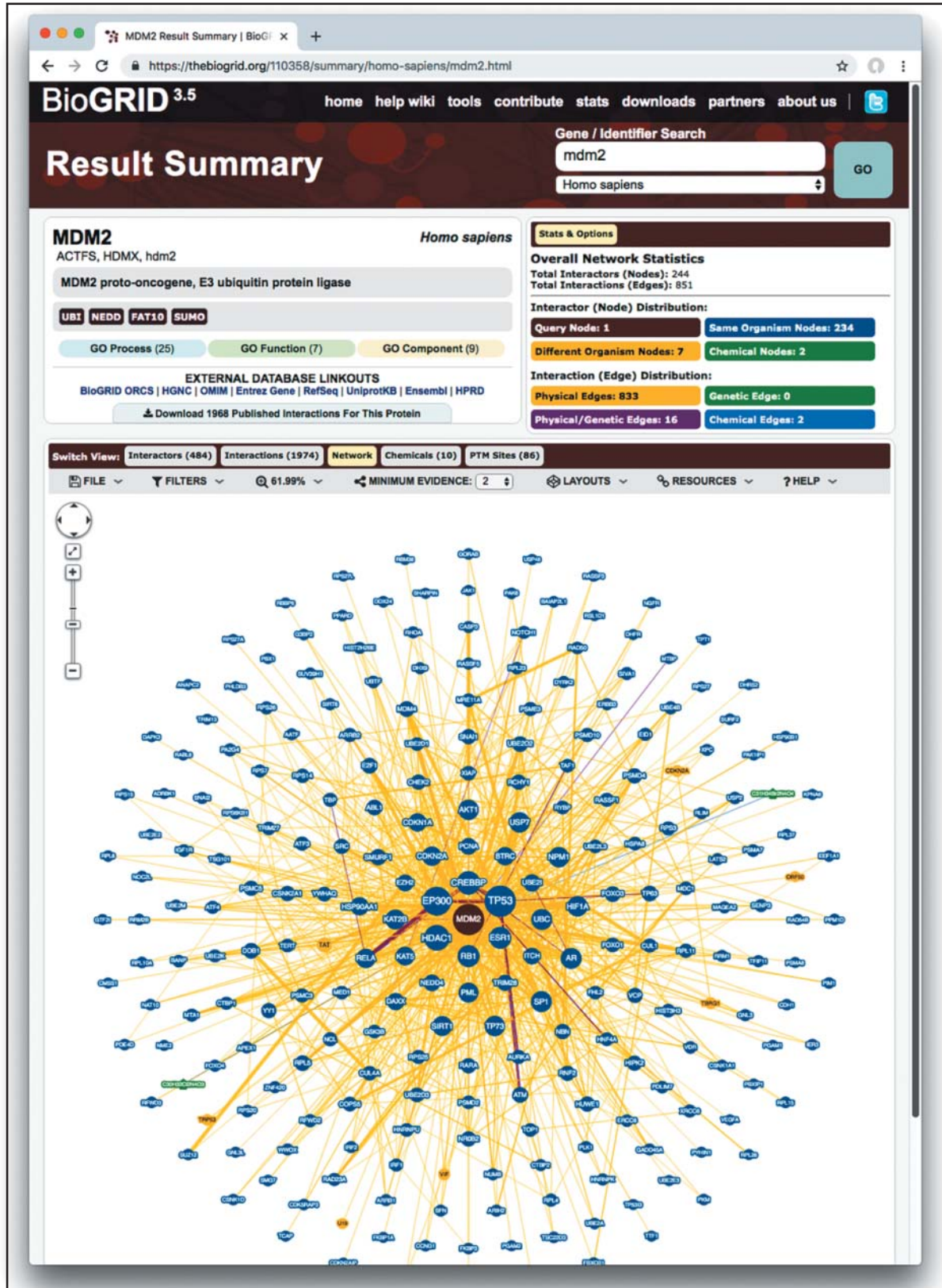


Figure 13.4 A BioGRID database record. A screenshot of the result page for a BioGRID search of the human *MDM2* gene. The top part of the page summarizes information about *MDM2*, including statistics about how many interactions BioGRID contains for this gene (top right). Five different views are available: interactors, showing a table of other genes that interact with *MDM2*; interactions, showing each experimental interaction evidence; a network view (shown in this figure); a table of chemical interactors; and a table of post-translational modifications.

EMBL-EBI Services Research Training About us

IntAct mdm2 Advanced
Examples: BRCA2, Q06609, dmc1, 10831611

Home Advanced Search About Resources Download Feedback

IntAct > IntAct Search Results + Show more data from EMBL-EBI

716 binary interactions found for search term *mdm2*

Interactions (716) Interactors Interaction Details Graph

Filter out the spoke expanded co-complexes (137) What is this view?

Your query also matches **11,875** interaction evidences from **10** other databases. (1 database(s) non responding)

Your query also matches **750** interaction evidences from **2** other IMEx databases.

Customize view Select format to Download Download

(1 of 36) 1 2 3 4 5 6 7 8 9 10 >> >

Dts	Molecule 'A'	Links 'A'	Molecule 'B'	Links 'B'	Interaction Detection Method	Interaction AC	Source Database
1	MDM2	Q00987 EBI-389668	TP53	P04637 EBI-366083	ubiquitinase assay	EBI-9350109 imex : IM-22307-12	IntAct
2					ubiquitinase assay	EBI-9350102 imex : IM-22307-11	IntAct
3					enzyme linked immunosorbent assay	EBI-9074450 imex : IM-21984-2	IntAct
4					enzyme linked immunosorbent assay	EBI-9074437 imex : IM-21984-1	IntAct
5					anti tag coimmunoprecipitation	EBI-15759080 imex : IM-15267-3 dip : DIP-120E	DIP
6					isothermal titration calorimetry	EBI-15762398 imex : IM-15024-4 dip : DIP-120E	DIP

Figure 13.5 An IntAct database search for the human *MDM2* gene. A summary of all interactions that mention *MDM2* is displayed as a table, along with information about matches found in other interaction databases. A table of interactors and a network view of the results can also be viewed. Details about each interaction can be accessed by clicking on the small magnifying glasses on the left side of each row, and information can be downloaded in standard formats using the toolbar at the top of the table.

STRING

The STRING resource (Szklarczyk et al. 2015) makes available a diverse range of functional and experimental interaction information for over 2000 genomes in a graphical and user-friendly manner. Interaction types provided include gene neighborhood, gene fusion, phylogenetic profile, co-expression, publication article, gene name co-mentions, and experimentally determined protein–protein interactions. STRING enables searching by gene name, accession number, and sequence of interest. Results are graphically displayed and scored using a STRING-specific scoring scheme that correlates with validated protein–protein interactions and known pathways. A unique feature of STRING is the ability to examine in detail each separate evidence source supporting an interaction. Figure 13.6 shows a screenshot of STRING results. All STRING functional interactions can be freely downloaded for local use.

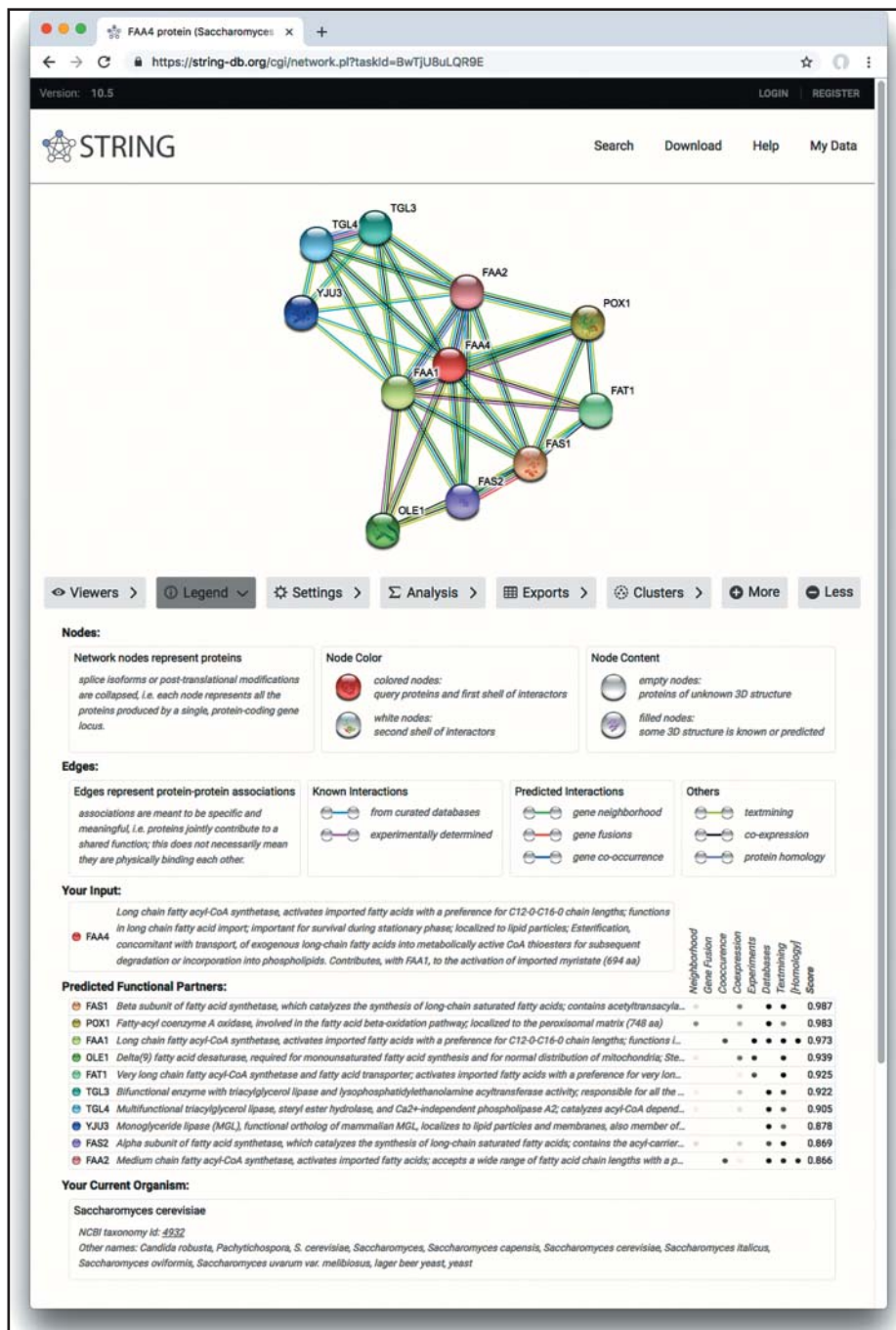


Figure 13.6 An example of the main STRING query result page. A network of relationships involving the query gene (here, budding yeast gene *FAA4*) and a set of functionally related genes, with different colored lines indicating which prediction or experimental method supports each link. The legend is provided graphically in the figure. The table at the bottom summarizes the overall strength of each interaction and the contribution to this score from the various evidence sources. Clicking the Viewers button at the far left provides access to dedicated graphical and textual reports for each evidence source. Pathway (and other gene set) enrichment is accessible by clicking on the Analysis button. The More and Less buttons under the network illustration increase or decrease the number of related genes shown.

GeneMANIA

GeneMANIA (Franz et al. 2018) is similar to STRING, but focuses on nine major model organisms and collects data from different sources than STRING. GeneMANIA also uses a search algorithm that computes the relevance of each functional interaction network based on the query and supports user-uploaded networks. Given a single query gene, GeneMANIA finds genes that likely share similar function based on their interactions. Given a set of genes, GeneMANIA finds functionally similar genes to the set, using a machine learning algorithm similar to that used by popular movie websites to recommend films a user would like, based on films they have previously watched. For example, if a set of kinases is queried, GeneMANIA will find similar kinases, upweighting the protein domain similarity network. If members of a pathway are queried, GeneMANIA will predict other members of that pathway, likely based on physical and co-expression interactions (Figure 13.7). All functional interactions found within GeneMANIA can be freely downloaded.

Strategies for Navigating Pathway and Interaction Databases

The number and diversity of pathway and interaction databases can be bewildering. From a general and practical perspective, users should start their searches using the above databases, since they are the largest and most actively developed resources available. Users should also be aware of meta-databases that collect information from multiple other databases providing a convenient single point of access. Examples of these kind of consolidated resources are Pathway Commons for pathways (Cerami et al. 2011), iRefIndex for protein interactions (Razick et al. 2008), and OmniPath for activation/inhibition interactions in signaling pathways (Turei et al. 2016), though many more exist. A good starting point to explore the wider set of pathway and interaction databases is the Pathguide link directory (Bader et al. 2006).

This section provides criteria that can be used to evaluate the quality and utility of a pathway and interaction database. The criteria include scope, data quality, “freshness” of data, data quantity, availability, and technical architecture, each of which we will cover in turn. The scope of a database (or what types of records it collects) is important to know prior to searching for information. For example, the BioGRID database contains information about protein–protein interactions and genetic interactions, two related data types with very different properties. It is possible for a user to search for protein interactions, only to find genetic interactions and subsequently misinterpret them as protein interactions if they are not aware of the database scope. Data quality depends heavily on level of curation and validation and can be difficult to independently assess. General things to look for are evidence of manual curation, which usually indicates higher quality data, versus databases that contain computationally predicted information that is not manually reviewed. While expert curated databases are the gold standard, collections of lower quality information are still useful but generally require that the user has the expertise and time to sort through it. For instance, databases of protein–protein interactions created automatically by literature extraction techniques (text mining) may only be 70% accurate but might still have some correct information that no other database contains. Data freshness is also important, and databases that are well maintained and updated regularly often indicate higher data quality. Users should look for dates on the homepage of the database as well as in the records, or creation times of datasets available on download sites when available, to find out how recent the data are. Another measure of the utility of a database is data quantity, where the more data available (assuming they are of good quality) the better. Users should also be aware of database availability, or licensing terms, as some databases have intellectual property restrictions. Fortunately, many databases are either freely available to all or free for use by academic researchers. Finally, if one plans to analyze a given database as a batch, the technical architecture of the database should be considered (Helmy et al. 2016). Ideally, the database will be available in standard formats (described in Standard Data Formats for Pathways and Molecular Interactions) and provide application programming interfaces (APIs). In

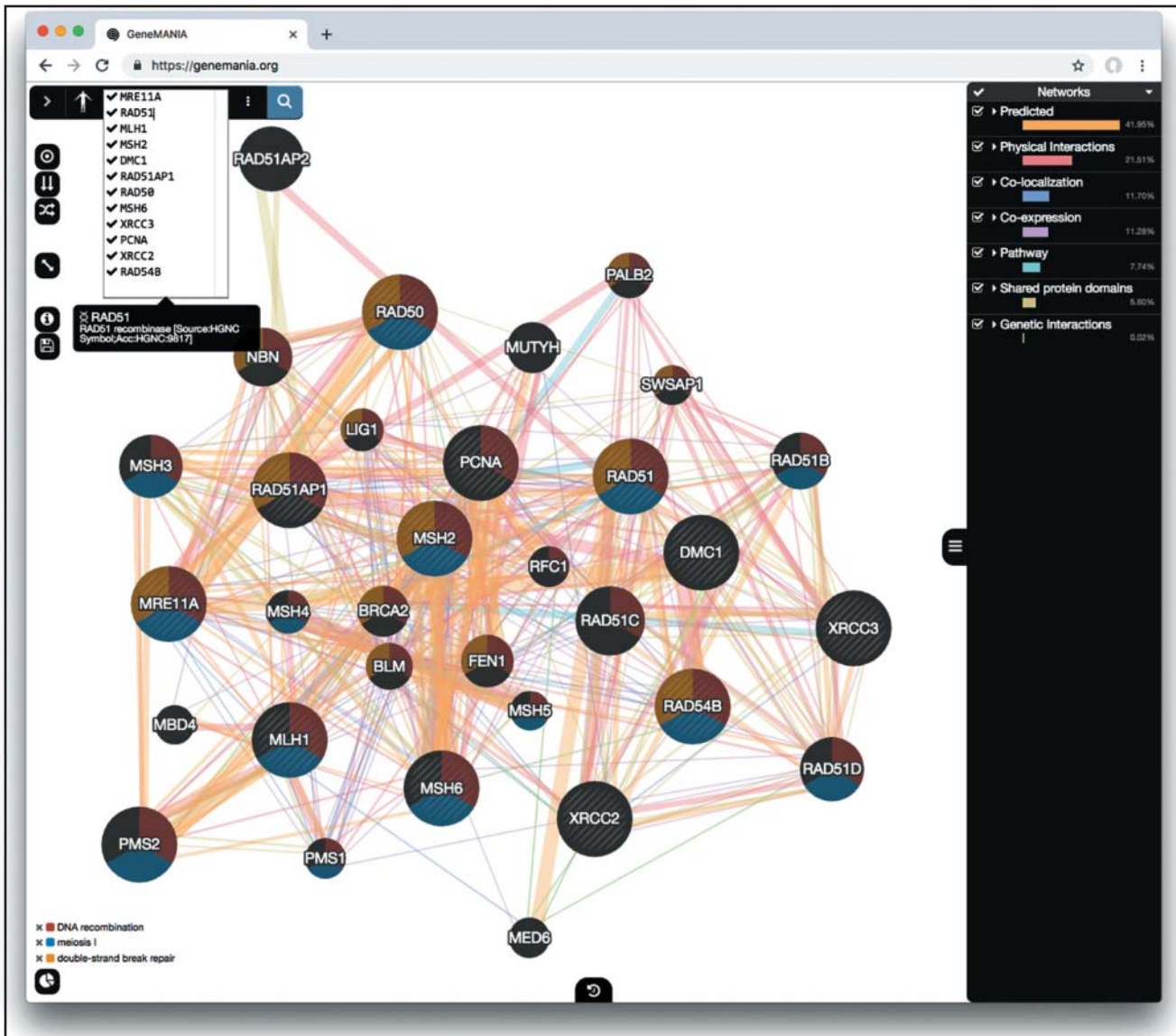


Figure 13.7 A query result from GeneMANIA. Each node in the network represents a gene, while the connections represent different types of functional interactions; these are color coded based on the key in the right panel (e.g. co-expression is purple). The example query for human (*MRE11A*, *RAD51*, *MLH1*, *MSH2*, *DMC1*, *RAD51AP1*, *RAD50*, *MSH6*, *XRCC3*, *PCNA*, *XRCC2*, *RAD54B*) was used to generate this network. Query nodes are shown with diagonal light gray lines. An additional 20 related genes (default parameter) are shown, with node size proportional to how strongly connected the related genes are to the query genes, with larger nodes being more strongly connected. Clicking on the pie chart at the bottom left opens a Gene Ontology annotation panel that enables a user to color nodes by selected pathways. Three pathways were selected to color nodes in this example, as shown in the legend at the bottom left.

summary, to get the most value out of pathway and network databases, it is important to study the database and understand how to properly use it.

Standard Data Formats for Pathways and Molecular Interactions

This section provides an overview of standard data exchange formats. Ideally, one would be able to access all relevant pathway and network information needed to solve a research problem from one convenient source. In reality, each database group creates its own way to represent data, making it extremely difficult to combine data and use them for comprehensive

analysis. Fortunately, standard data formats have been developed that many databases support and that make it easy to access data from diverse sources in one (or a small number of) compatible formats.

BioPAX

The Biological Pathway Exchange (BioPAX) format is a standard language to represent biological pathways (Demir et al. 2010). BioPAX can represent metabolic and signaling pathways, molecular and genetic interactions, and gene regulation networks (Figure 13.8). BioPAX is written in the Web Ontology Language (OWL), which is an XML language that can capture classes, class properties, and their relationships. The top-level class in BioPAX is Entity, which encompasses four types: Pathway, Interaction, Gene, and PhysicalEntity. A Pathway is a collection of Interactions, optionally ordered in steps. Interactions contain genes or physical entities: protein, DNA, RNA, small molecule, and complex. There are four major types of representation styles and data types covered. Biochemical and signaling

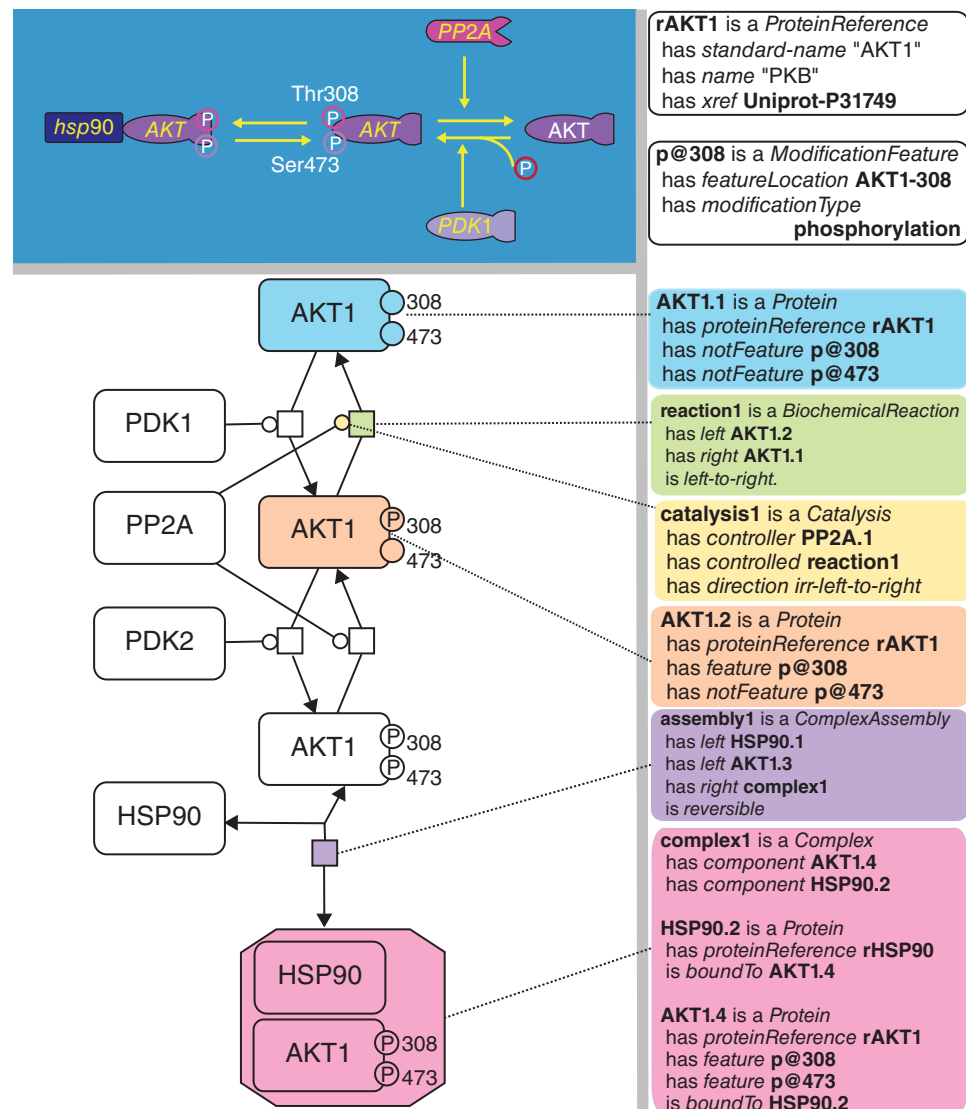


Figure 13.8 The AKT pathway as represented by a traditional method (top left, from www.biocarta.com), a formalized SBGN diagram (at left, from www.sbgng.org), and using the BioPAX language (on the right). Source: Reproduced with permission of Springer from Demir et al. (2010).

pathways consist of conversion and control interaction types, with physical entity participants that maintain state (e.g. post-translational modifications, cell location, and protein complex bound). Conversion captures biochemical reactions, transport, degradation, and complex assembly. Control captures catalysis and modulation. Catalysis describes the enzyme that catalyzes a biochemical reaction, so two interactions are required to capture a step in a metabolic pathway (catalysis and biochemical reaction). Gene regulatory pathways consist of TemplateReaction and TemplateReactionRegulation interactions. A template reaction is one that converts between molecules in the central dogma, such as DNA to RNA, or DNA to protein. A TemplateReactionRegulation type of interaction controls this, capturing, for instance, a transcription factor that regulates the expression of a gene (described as a TemplateReaction). The MolecularInteraction class captures protein–protein or other molecular interactions and follows the style of the PSI-MI standard (see PSI-MI). The GeneticInteraction class represents genetic interactions such as synthetic lethal or epistatic interactions between genes. Many databases make their data available in BioPAX, including Reactome (Fabregat et al. 2018), the BioCyc database family (Caspi et al. 2018), and Pathway Commons (Cerami et al. 2011). BioPAX provides a programming library to support software developers loading, saving, and querying BioPAX files (Demir et al. 2013) and provides a validator service (Rodchenkov et al. 2013) to support content providers creating new BioPAX files.

PSI-MI

PSI has developed an XML-based format for exchanging protein–protein interactions, called PSI-MI (Hermjakob et al. 2004). The data model of the format contains an “interaction” record comprising a set of proteins that interact (could be more than two), an “experimental conditions” controlled vocabulary, and information about publication references and protein features, such as binding sites and post-translational modification sites. The PSI-MI group also maintains an extensive ontology of terms describing concepts such as interaction and experimental method types that is used as a controlled vocabulary throughout PSI-MI (Mayer et al. 2014). Figure 13.9 shows the top level of a PSI-MI record. Many databases and tools support PSI-MI, and, similar to BioPAX, software libraries, web services, and a validator are available to aid software and database groups to support the standard. The PSI-MI and BioPAX developer groups worked together to ensure compatibility between formats so that the MolecularInteraction class in BioPAX is interconvertible to and from PSI-MI. PSI-MI also makes available a tab-delimited version of the format called MITAB (molecular interaction tab delimited) that makes it easier to process the files in scripts.

SBML

The Systems Biology Markup Language (SBML) is an XML-based format for exchanging mathematical pathway simulation models (Hucka et al. 2003). An example of a mathematical pathway model is a system of ordinary differential equations that describe the rates of all of the reactions in a pathway. With the right parameters (for example, initial concentrations of molecules and kinetic constants of reactions) the computer can calculate the concentration of the various molecular species in a pathway over time. A number of simulation tools support these formats. The BioModels database contains SBML models for many pathways (Le Novere et al. 2006) and many software tools are available to simulate SBML models.

Pathway Visualization and Analysis

Pathway visualization tools are computer programs that can automatically draw a pathway diagram. Automated pathway visualization tools, especially for browsing metabolic pathways, have been around since soon after the first metabolic databases were built. For instance, a

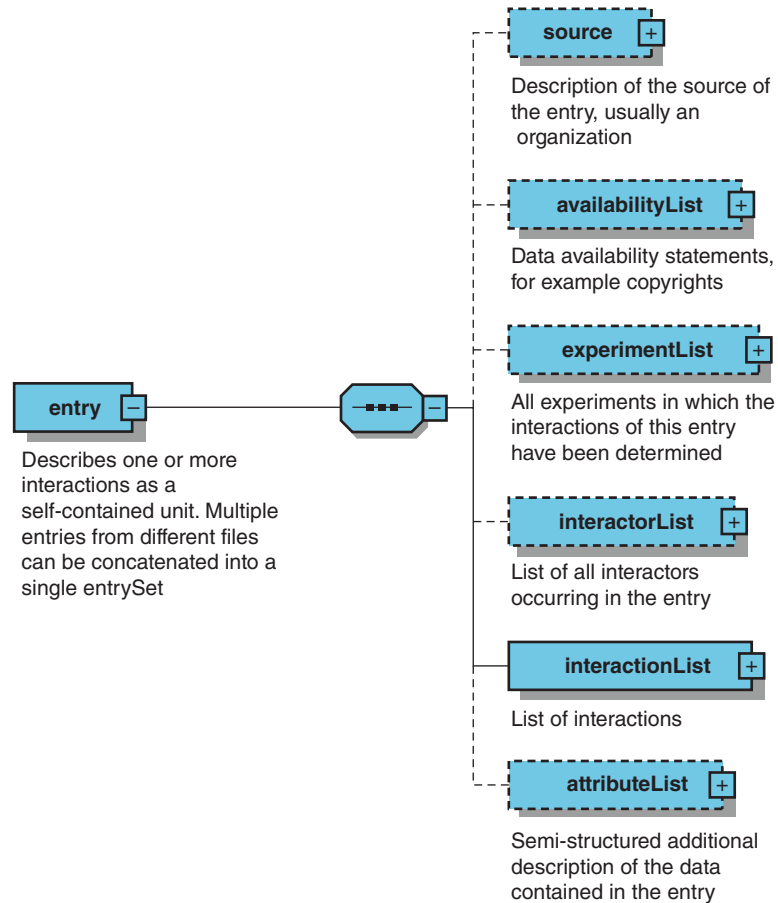


Figure 13.9 The main components of the Proteomics Standards Initiative–Molecular Interactions (PSI-MI) data model for describing protein–protein interactions. Boxes represent defined XML data types. Dashed lines represent optional elements. The hexagonal box represents a collection of elements that are below it. Minus and plus symbols in small boxes represent expanded and collapsed views of each element, respectively. Collapsed boxes have more elements inside them that are not shown. The full schema for PSI-MI is on the PSI-MI Web site.

pathway-drawing tool is present in the ACeDB database (Eeckman and Durbin 1995) and in EcoCyc (Karp et al. 2002). Many of these tools display static pictures with components, such as enzymes or small molecules, that can be clicked on to get more information about the component from a source database. Examples of static clickable pathway images can be found in the Reactome database (Figure 13.1). More advanced tools are able to dynamically generate pathway diagrams from an underlying database that allow the user to change how the pathway is viewed. For instance, the EcoCyc database contains a pathway visualization tool that can display varying levels of detail about a pathway, from an overview to a detailed view showing all chemical structures of small molecules in the pathway (Figure 13.10). PathVisio can display pathway diagrams from multiple sources and can aid interpretation of gene expression and other genomics data by overlaying them on a pathway diagram (Figure 13.11; Kutmon et al. 2015). Generically, PathVisio and similar tools must be able to load pathway information and gene expression data and match genes from one dataset to the other. Usually this requires gene identifiers to match between the two datasets, though many tools provide features to help map identifiers from one type to another to help match genes between datasets, but this can sometimes be error prone (Zeeberg et al. 2004).

The Systems Biology Graphical Notation (SBGN) is a standard format for pathway diagrams (Le Novère et al. 2009). Three versions exist to capture different pathway representation paradigms. Process Description (PD) diagrams visualize biochemical-style metabolic and

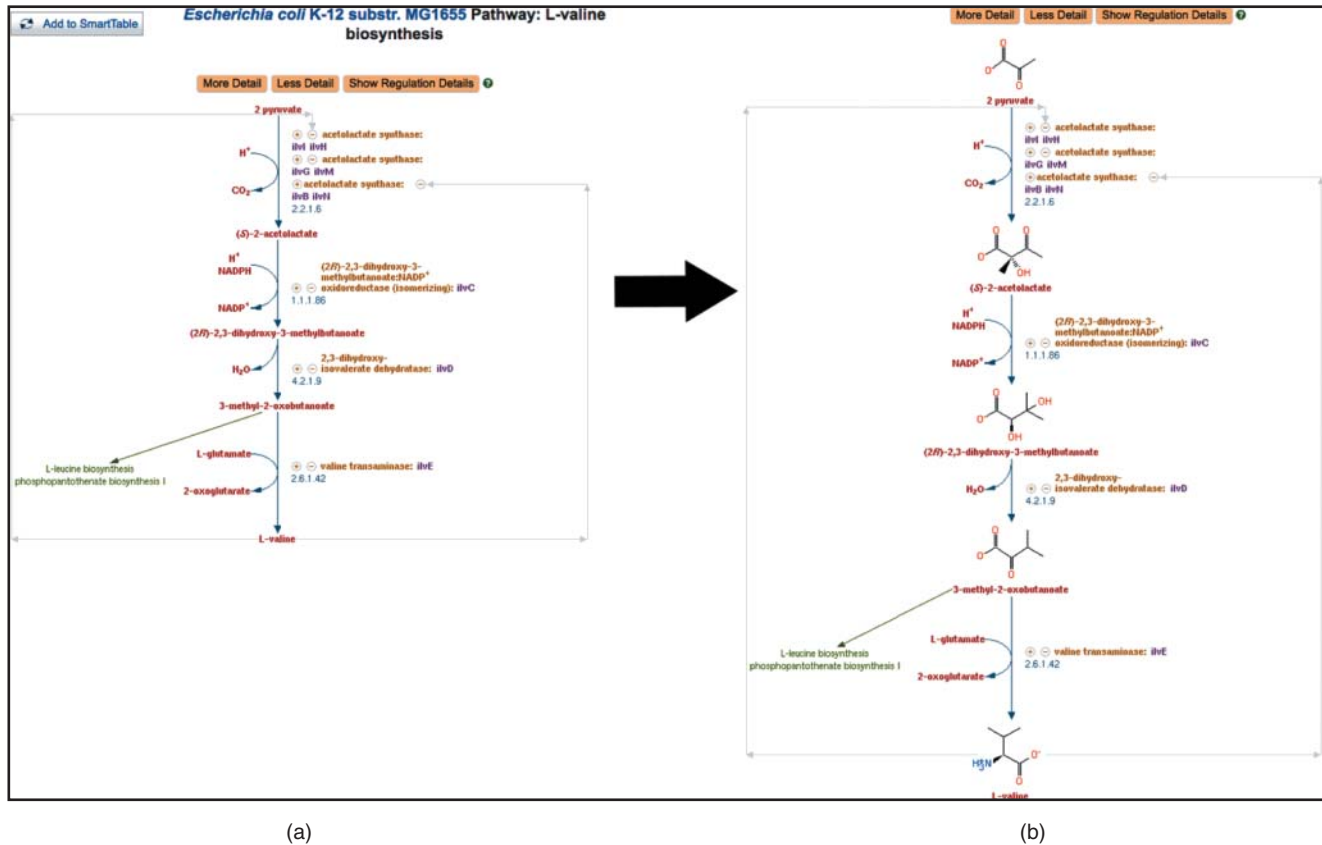


Figure 13.10 The valine biosynthesis pathway dynamically drawn by the Pathway Tools software that supports the BioCyc family of databases. The advantage of automatic pathway diagram layout is that the diagram can be drawn according to user preference. Here two views of the same pathway are shown, the one in (b) providing more detail than the one in (a). Notice the presence of small molecule structures in (b). Nodes in the pathway diagram represent metabolites and connections represent enzymes.

signaling pathways (Figures 13.12 and 13.13). Entity Relationship (ER) diagrams display an interaction network involving participants. Activity Flow (AF) diagrams show how information flows within a pathway, including activation and inhibition relationships. SBGN-ML is a standard XML format for exchanging SBGN diagrams (van Iersel et al. 2012) and many editors and visualization tools support automatically drawing SBGN diagrams (Sari et al. 2015; Hartmann and Jozefowicz 2018).

The major type of pathway analysis method is pathway enrichment analysis, which is used to interpret genomics and other genome-scale data. It identifies pathways that are more or less enriched than expected in a large gene list, typically derived from high-throughput transcriptomic or proteomic methods. In this analysis, pathways are statistically tested for over-representation in the experimental gene list above what is expected by chance. For instance, an experimentally derived gene list containing 50% cell cycle genes is surprisingly enriched given that only 8% of human protein-coding genes are involved in this process.

Pathway enrichment analysis involves three major steps. First, one must define a gene list of interest using available high-throughput data. Raw data from such an experiment generally require computational processing, such as normalization and scoring to identify genes of interest. For example, a list of genes differentially expressed between two groups of samples can be derived from RNA-seq data. Second, pathway enrichment analysis is performed. A statistical method is used to identify pathways enriched in the gene list from the first step, relative to what is expected by chance. All pathways in a given database are tested for enrichment in the gene list, and the resulting *p* values are corrected for multiple hypothesis testing to

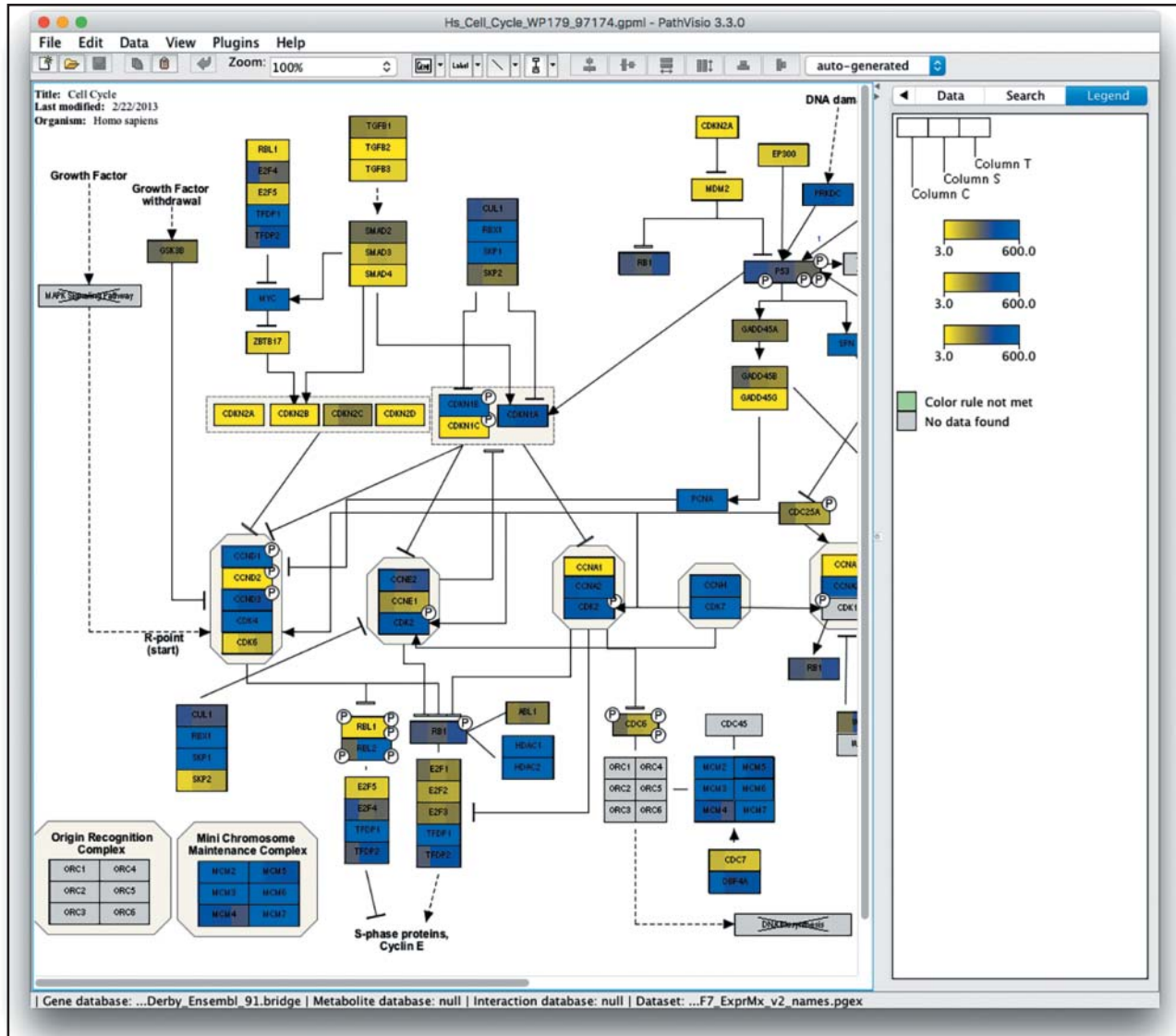


Figure 13.11 Output from the PathVisio software showing a portion of a human cell cycle pathway overlaid with gene expression data from three breast cancer cell line samples. Rectangles represent genes or pathways, as labeled, and are colored based on gene expression levels according to the legend on the right, where yellow represents low mRNA gene expression and blue represents high mRNA gene expression.

identify significantly enriched pathways. Third, pathway enrichment analysis results are visualized and interpreted. Many enriched pathways may be identified in the second step, often including related versions of the same pathway. Visualization can help identify the main biological themes and their relationships in this list for focused study.

Many statistical methods have been proposed to perform pathway enrichment analysis (Khatri et al. 2012), though there are two major types that are tailored for specific types of gene lists. The first is designed to analyze a gene list containing tens to several thousands of genes, as may be defined in the course of a cancer genomics experiment (i.e. the set of all genes mutated in a cancerous sample as compared with a normal sample). This type of gene list can be analyzed using a Fisher's exact test to calculate the probability of a non-random association between genes in the input list and those in a pathway. This test is repeated for all pathway gene sets in a database, correcting for repeated tests (multiple hypotheses) using the Benjamini–Hochberg false discovery rate (FDR) method (Hochberg and Benjamini 1990). The result is a set of pathway gene sets significantly enriched in a gene list and their associated

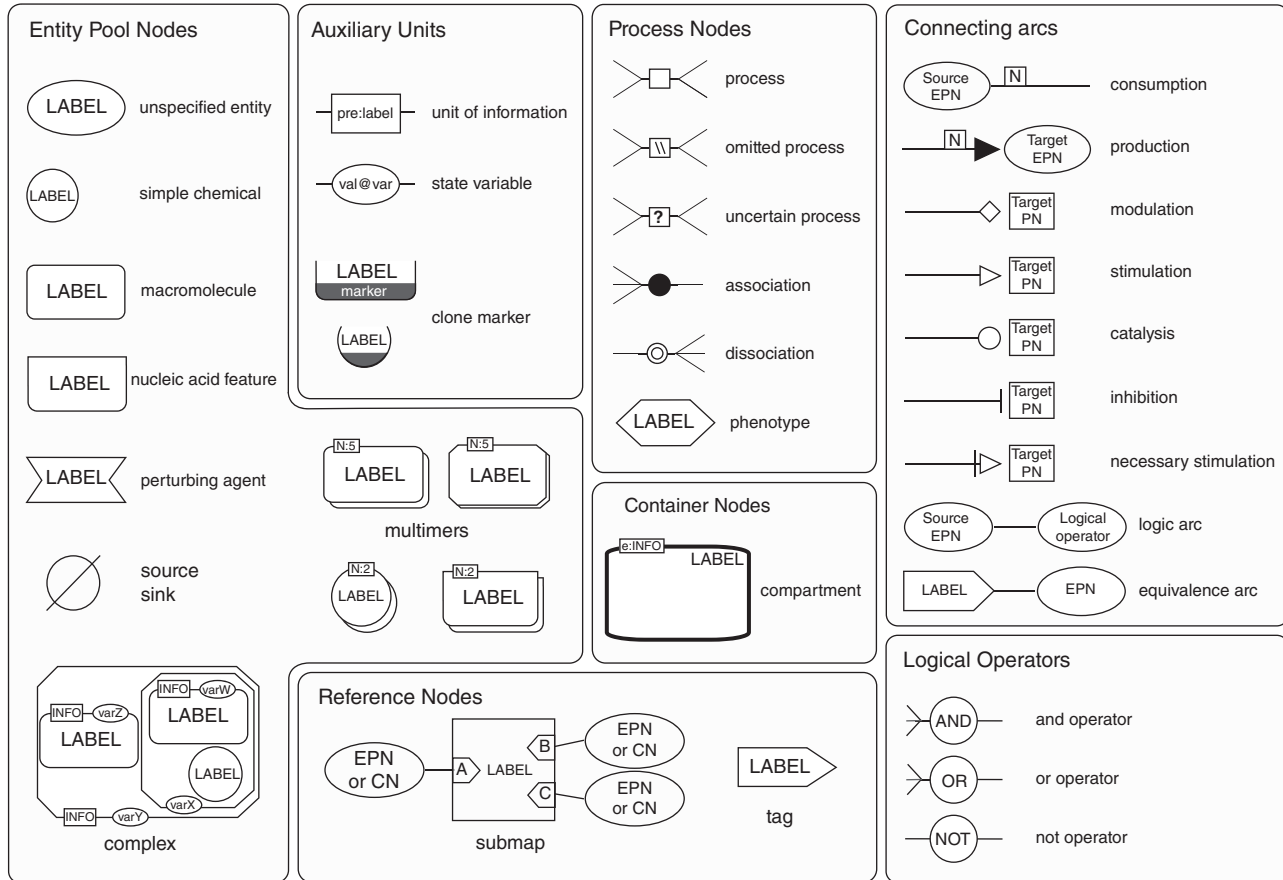


Figure 13.12 The set of symbol types available in the Systems Biology Graphical Notation (SBGN) Process Description (PD) format.

corrected p values (also called q values). A useful tool to perform this analysis is the web-based g:Profiler system (Figure 13.14; Reimand et al. 2016).

A second important type of gene list is ranked by an experimental score. A typical example is the list of all differentially expressed genes in the genome measured in an RNA-seq experiment comparing one condition with another. This gene list is ranked by a differential expression score, with most positively differentially expressed (upregulated) genes in condition A vs. B at the top of the list, genes not differentially expressed in the middle of the list, and genes negatively differentially expressed (downregulated) at the bottom of the list. There is usually no natural way to threshold this list to define a smaller list of genes suitable for input into Fisher's exact test-based analysis methods. Further, thresholding may remove biologically relevant signal, as genes that are weakly differentially expressed may contribute signal to a given enriched pathway. To address this, rank-based pathway enrichment analysis methods have been developed; these methods do not require a threshold to be defined and, instead, consider all genes in the list. The most commonly used method of this type is Gene Set Enrichment Analysis (GSEA), primarily implemented as free software available for local installation on a desktop (Subramanian et al. 2005). The GSEA method searches for pathways whose genes are enriched at the top or bottom of the ranked gene list, more so than expected by chance. For instance, if the top most differentially expressed genes are involved in the cell cycle, this suggests that the cell cycle pathway is regulated in the experiment. In contrast, if cell cycle genes appear randomly scattered through the whole ranked list, the cell cycle pathway is likely not significantly regulated. To calculate an enrichment score (ES) for a pathway, GSEA progressively examines genes from the top to the bottom of the ranked list, increasing the enrichment score if a gene is part of the pathway and decreasing the score otherwise. These running sum values are weighted, so that enrichment of the top- (and bottom-) ranking genes is amplified,

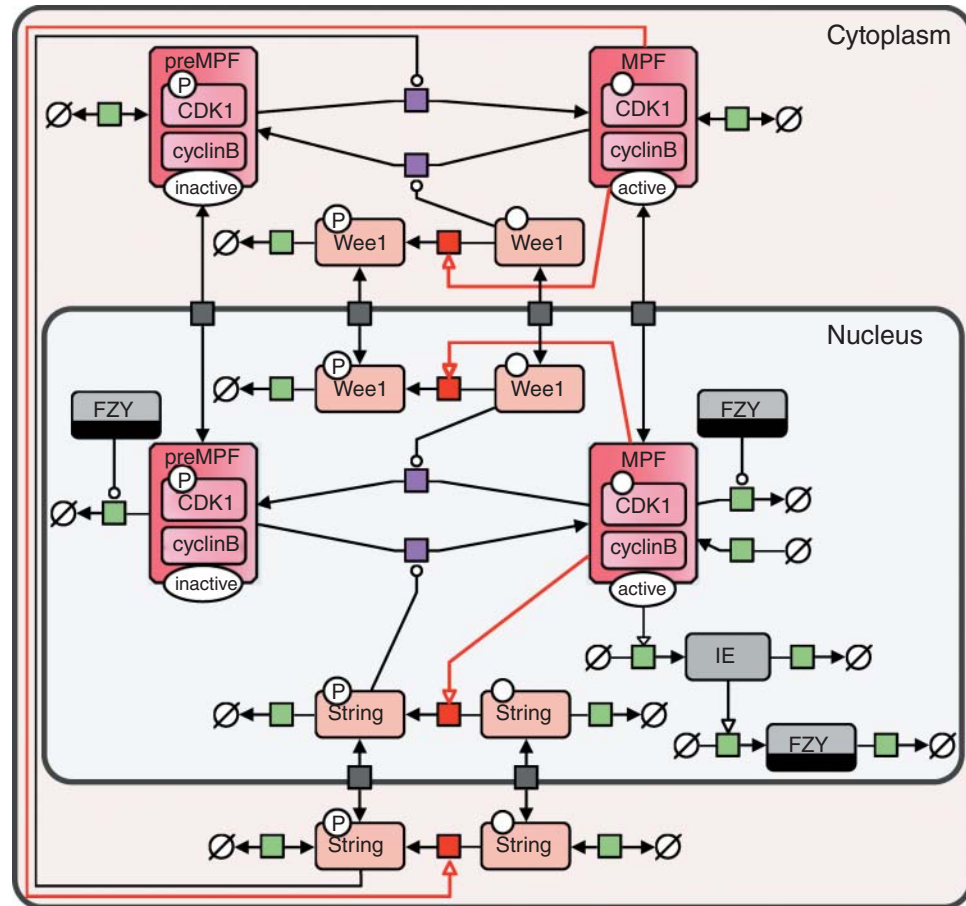


Figure 13.13 The *Drosophila melanogaster* cell cycle drawn using Systems Biology Graphical Notation (SBGN) Process Description (PD) and colored to increase visual appeal. Source: Reproduced from Toure et al. (2018).

whereas enrichment in genes with more moderate ranks is not amplified. The ES is calculated as the maximum value of the running sum and normalized relative to pathway size, resulting in a normalized enrichment score (NES) that reflects the enrichment of the pathway in the gene list. Positive and negative NES values represent enrichment at the top and bottom of the list, respectively (Figure 13.15). This process is repeated for each pathway in a database. Finally, a permutation-based p value is computed and corrected for multiple testing to produce a permutation-based FDR q value that ranges from 0 (highly significant) to 1 (not significant). Permutation p values are computed either by repeating the analysis many times with random gene sets or with random assignment of experimental class labels, like “case” and “control,” with the latter recommended if more than five samples are available. The same analysis is performed starting from the bottom of the ranked gene list to identify pathways enriched in the bottom of the list. Resulting pathways are selected using the FDR q value threshold (e.g. $q < 0.05$) and ranked using NES. It is also useful to inspect the “leading edge” genes that contribute to the increase in the enrichment score before it peaks.

In both of the above-described analysis types, significantly enriched pathways are typically displayed as a table. Pathway information is inherently redundant, as genes often participate in multiple pathways and pathways collected from different databases may be repeated. Pathway enrichment analysis often highlights several versions of the same pathway as a result. Collapsing redundant pathways into a single biological theme simplifies interpretation. The Enrichment Map visualization software is an app within the Cytoscape network visualization and analysis software (described in Network Visualization) that addresses this problem (Bindea et al. 2009; Merico et al. 2010). An enrichment map is a network

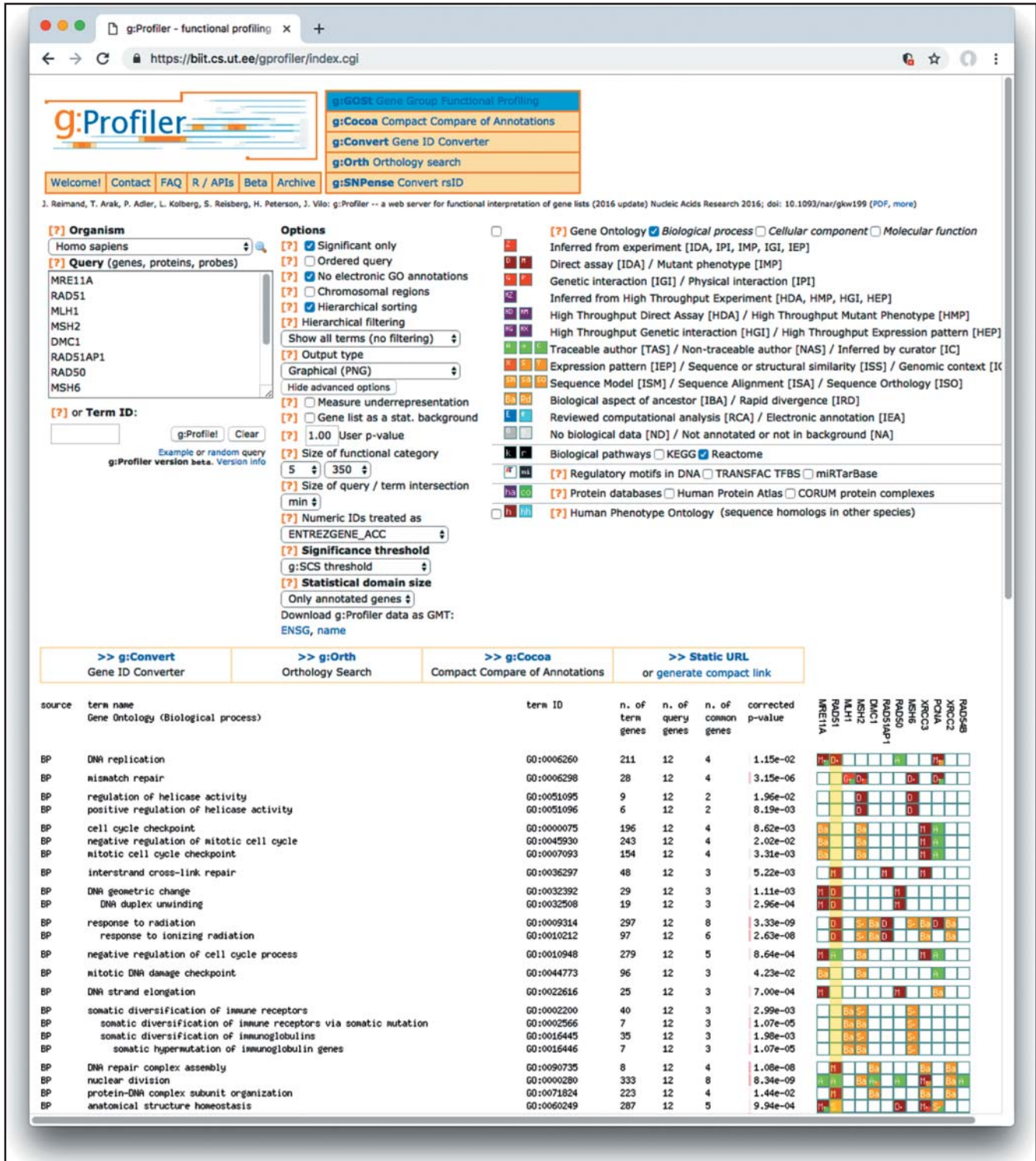


Figure 13.14 The results of pathway enrichment analysis using the g:Profiler tool. The top part of the screen shows the input form, including a text box to enter a gene list (top left), various analysis options to select (top center), and gene set databases that can be selected (top right). The enrichment results are shown in a table at the bottom of the figure. Each row includes the pathway name (left column), enrichment statistics (center columns), and a graphical view of the query genes and which pathways they belong to (right graphic).

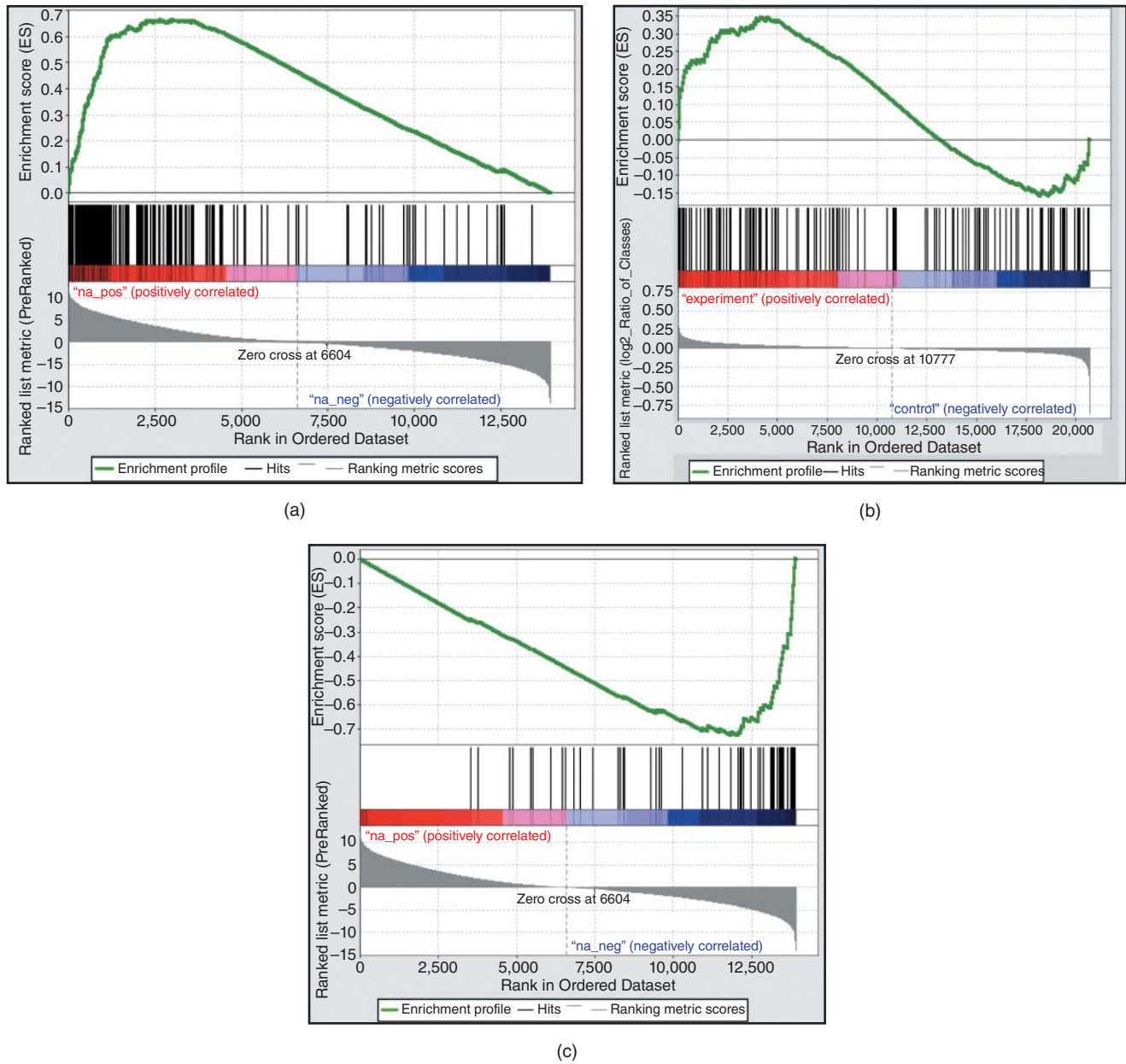


Figure 13.15 A Gene Set Enrichment Analysis (GSEA) enrichment figure. The bottom half of the figure represents the full ranked list of genes, ranked from high (left) to low (right). Genes in the ranked list that match a pathway (gene set) are shown as black vertical lines. The running enrichment score is plotted in green at the top. This is an example of a highly enriched pathway, as the green line quickly rises to a high level before decreasing. Three figures are shown, with good enrichment in the top part of the ranked gene list (a), poor enrichment (b, with a random spread of pathway genes across the ranked list), and good enrichment at the bottom of the list (c).

representing overlaps among enriched pathways (Figure 13.16). Pathways are represented as circles (nodes) that are colored by enrichment score and are connected with lines (edges) sized based on the number of genes shared by the connected pathways. Network layout and clustering algorithms are used to automatically display and group similar pathways as major biological themes. Interactive exploration of pathway enrichment score (filtering nodes) and connections between pathways (filtering edges) is possible. If the gene expression data are optionally loaded, clicking on a pathway node will display a gene expression heat map of all genes in the pathway. Multiple enrichment analysis results can be simultaneously visualized in a single enrichment map to enable comparison, in which case different colors are used on the nodes for each enrichment result (Reimand et al. 2019).

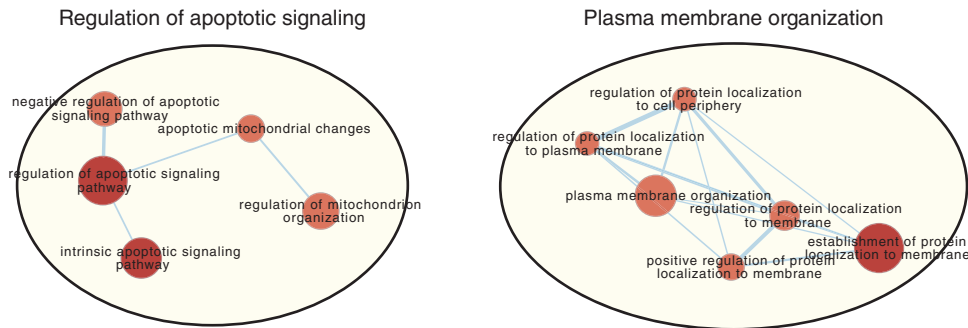


Figure 13.16 An enrichment map showing two enriched themes. Each node represents a pathway gene set, named as labeled. The size of the node is proportional to the number of genes in the pathway. The color of the node represents the enrichment score, with a better score indicated by a deeper shade of red. Edges represent genes shared between two pathways, with thicker edges indicating more shared genes. Related pathways have been automatically grouped into themes shown by large labeled and shaded circles using Cytoscape's AutoAnnotate app.

Network Visualization and Analysis

While viewing individual pathways is useful for detailed mechanistic studies, it is not amenable to visualization and analysis of large sets of molecular interactions and knowledge mapped outside of well-studied pathways. For this reason, network visualization and analysis tools have been developed. Network visualization and analysis relies on concepts from the computer science field of graph theory, so we begin this section with a brief discussion of basic graph theory concepts. Graph theory is based on the notion of a graph, a representation of connected data as a set of nodes (or vertices) and a set of connecting edges (Figure 13.17). Edges may be directed, in which case they may be called arcs. Nodes and edges may have associated weights or other data values. Different classes of graphs exist; for instance, a graph that does not contain any cycles is called acyclic (also called a tree). Tree graphs have a root node and leaf nodes, and a collection of trees is termed a forest. An example of a directed acyclic graph in bioinformatics is the Gene Ontology (see Chapter 7; Ashburner et al. 2000), where the most general annotation term is the root and the most specific terms are leaf nodes. The number of edges connected to a node in an undirected graph is called the degree. For a directed graph, the in-degree and out-degree are the number of arcs input and output from a node, respectively. A graph is an abstract mathematical concept and can be mapped to any problem where a mapping can be imagined; thus, direction, weight, and connectivity do not have any specific biological (or other domain) meaning until a mapping is made.

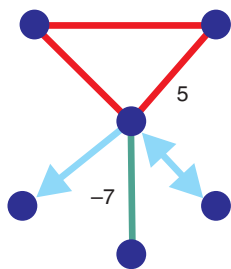


Figure 13.17 An introduction to terminology and visual notation used in the computer science field of graph theory. Blue circles are nodes or vertices (singular is vertex), undirected lines (red and green) are called edges, and directed lines (cyan) are called arcs. Nodes or edges can have associated attributes, such as weights. Here, two edge weights are shown: 5 and -7 . A series of edges that form a closed loop is called a cycle (red lines). The colors are present in this figure solely to annotate the graph and are not part of normal visual notation. A graph is an abstract mathematical concept. Edge direction, weights, and other attributes do not mean anything until mapped to a specific problem.

Intuitively, biomolecular interaction networks can be mapped to a graph, where biomolecules are represented as nodes and interactions as edges. Other information could also be mapped; for instance, edge direction may represent activation relationships and edge weight may be mapped from confidence information about an interaction. Some types of biological interaction information cannot be faithfully mapped to a graph, or there may be multiple ambiguous mappings or mappings that cause loss of information. For example, protein complexes larger than two molecules detected in a co-immunoprecipitation experiment cannot easily be described using the binary relationships in a graph; rather, they can only be accurately represented as a set, since the direct physical connections between the proteins in the complex are not known from the experiment. The set can be mapped to a graph in different ways, such as by connecting all proteins in the set in a clique (a fully connected graph) or by creating a node that represents the set and linking each protein to the new node (Bader and Hogue 2002).

The reason graph theory is used to represent biological networks is that it is useful for answering many interesting biological questions. For instance, if one wants to find out if one protein connects to another protein in a protein interaction network, an algorithm (called a breadth-first search) can be run that is mathematically guaranteed to find the shortest path between the two nodes, if it exists. Many other useful graph algorithms exist to manipulate, query, analyze, and visualize graphs. More information about graph theory can be found in Box 13.1 and in books devoted to graph theory algorithms (Bollobás 1998; Mehlhorn and Näher 1999; Cormen 2001).

Box 13.1 Advanced Graph Theory Applications

There is a natural relationship between graph theory and linear algebra. Any graph can be represented as an $N \times N$ matrix, called an adjacency matrix, where the rows and columns represent the nodes in a graph and a “1” is placed in the matrix at position (i,j) if node i connects with node j . If the edges in the graph are weighted, the weight can be recorded at position (i,j) instead of using a “1.” Since many types of matrices in bioinformatics are $N \times N$, or square, they can be represented as a graph, and it is sometimes useful to make this conversion to visualize the matrix. One interesting example is a protein sequence similarity matrix, which records the sequence similarity (e.g. as calculated by BLAST (see Chapter 3)) of a set of sequences in an all-against-all fashion. The rows and columns of a similarity matrix represent the set of things being compared; in this case, protein sequences and matrix position (i,j) record the similarity score of protein i compared with protein j . By visualizing these data as a network instead of a matrix, the connections between clusters of similar proteins are more visually apparent (Akiva et al. 2014).

Mathematicians may also convert a graph to an adjacency matrix to apply algebraic matrix operations to the matrix to solve specific graph problems. Sometimes, the matrix operations are faster than the same operations performed directly on a graph using a standard algorithm. For instance, the entries (i,j) in the square of an adjacency matrix correspond to the number of paths of length 2 that exist in the graph between nodes i and j . This can be extended to higher powers of the adjacency matrix. Squaring the matrix quickly gives this answer if the matrix is sparse (filled with many zeros), but not as quickly if the graph is dense. Fortunately, many problems in biology translate to sparse graphs. One algorithm in bioinformatics that uses this mathematical problem-solving tactic to cluster a similarity matrix is the Markov cluster (MCL) algorithm (Enright et al. 2002). Through a series of adjacency matrix multiplications of the similarity graph and other mathematical operations, clusters of similar proteins are detected. Proteins in a similarity cluster have more paths between them than to proteins in other clusters. The matrix squaring operations are involved in counting the number of paths from one protein to another.

Network Visualization

Network visualization tools rely on algorithms from the computer science field of network layout. Typically, network layout algorithms try to make a graph look esthetically pleasing; that is, they try to minimize the overlap of nodes and crossing of edges so that as much of the graph as possible is clearly visible. Network layout is practical and generally works well on small- to medium-sized networks, such as those up to a few thousand nodes for a typically sized viewing area, such as a computer monitor. Larger networks than this require a larger than normal viewing area or the network to be reduced in complexity to view, such as by filtering edges or by zooming in to nodes of interest.

There are many types of network layout algorithms, such as arranging nodes hierarchically, in a circular fashion or in less structured formats. Importantly, the type of layout algorithm that will work best depends on the type of network that is input. For instance, a highly connected network will not display well when laid out hierarchically; only a truly hierarchical network, like a tree, will lay out well in this case. Thus, network visualization tools contain multiple layout methods that should all be tried to see which one generates an esthetically pleasing layout for a particular network.

One of the most commonly used layout types is called a spring-embedded algorithm, derived from the general class of force-directed layout algorithms and containing many variations. In a typical case, the network is modeled as a physical system where edges are springs and nodes are like-charged particles that repel. The layout starts by placing all nodes randomly and then calculates the position of each node given that long edges are like stretched springs and will pull the connected nodes close together, while nodes will repel each other the closer they get. By iterating over time, the network can stabilize on the final layout, which will have relatively short edges and relatively non-overlapping nodes. Think of this as taking a bunch of like-charged beads (nodes) connected by springs (edges), throwing them up in the air, and seeing what pattern they arrange themselves in when they land.

Once a network is laid out, it must be interpreted. There are three major patterns to look for in biological interaction networks (Merico et al. 2009). The first pattern takes the form of “guilt by association,” describing the phenomenon that genes of similar function are usually connected to each other in a protein or gene interaction network. This is useful for predicting the function of unknown genes based on the function of neighboring genes. The second pattern presents densely connected regions, or clusters, that frequently indicate pathways, systems, or molecular complexes. The third analysis pattern to examine are global features, such as how the densely connected regions are organized relative to each other, which may be helpful in understanding which ones are closely related.

Cytoscape is a freely available, open-source Java-based network visualization and analysis tool and is the most widely used tool of its type (Shannon et al. 2003). Cytoscape is able to visualize and analyze network data in the context of other types of data (e.g. genomic data) and lay out the network. Cytoscape networks are interactive and can be edited; nodes can be selected, dragged, and rotated using the mouse. Sophisticated node and edge selections can also be made by filtering based on user-defined combinations of loaded attributes and network topology. A major strength of Cytoscape is the ability to add new features by downloading “apps” from the Cytoscape app store (Lotia et al. 2013). Apps can be developed by anyone using the Java programming language. Hundreds of apps implementing a wide range of visualization and analysis methods have been developed and contributed to the project. Automating Cytoscape functions using R, Python, or other scripting languages is also possible (Demchak et al. 2018).

Cytoscape uses the concepts of network attributes and visual attributes when integrating and visualizing information on the network. There are two types of network attributes: node and edge. A node attribute is a data value that is associated with a node (usually by loading it from a file). If the node represents a protein, a node attribute could be the name of the protein, a term that describes the functional classification of that protein (perhaps from the Gene Ontology), or a protein abundance measurement. Similarly, an edge attribute is a data value that is associated

with an edge. If the edge represents an interaction among two proteins, an edge attribute could be the strength of the interaction or the type of experimental method that was used to detect the interaction. Multiple types of node and edge attributes can be loaded simultaneously, as long as each type has a different name. Either attribute can be discrete or continuous. An example of a discrete edge attribute is a list of interaction detection experimental methods. An example of a continuous node attribute is a set of gene expression values that range from 0.0 to 1.0.

Visual attributes in Cytoscape are aspects of a network diagram that can be displayed in different ways (Figure 13.18). These include shape, size, label, font, color, border color, and border type for nodes and label, font, color, line type (e.g. solid or dashed line), target arrow, and source arrow for edges. Once a network is loaded into Cytoscape, any node or edge attributes can be mapped to visual attributes using the Cytoscape visualization mapper, or “Style” system. A specific example of a visual style for a protein interaction network would be one that maps node attributes containing normalized gene expression values (ranging from 0.0 to 1.0, with 1.0 being the highest gene expression values in the set) to node color with an expression value of 0.0 mapping to green and 1.0 to red. Cytoscape will then automatically color all nodes continuously according to the style, and an expression value of 0.5 will be colored midway between green and red.

Cytoscape has hundreds of features that are documented in the Cytoscape manual, in online tutorials, and in various protocols, as well as through mailing lists involving a large community of users.

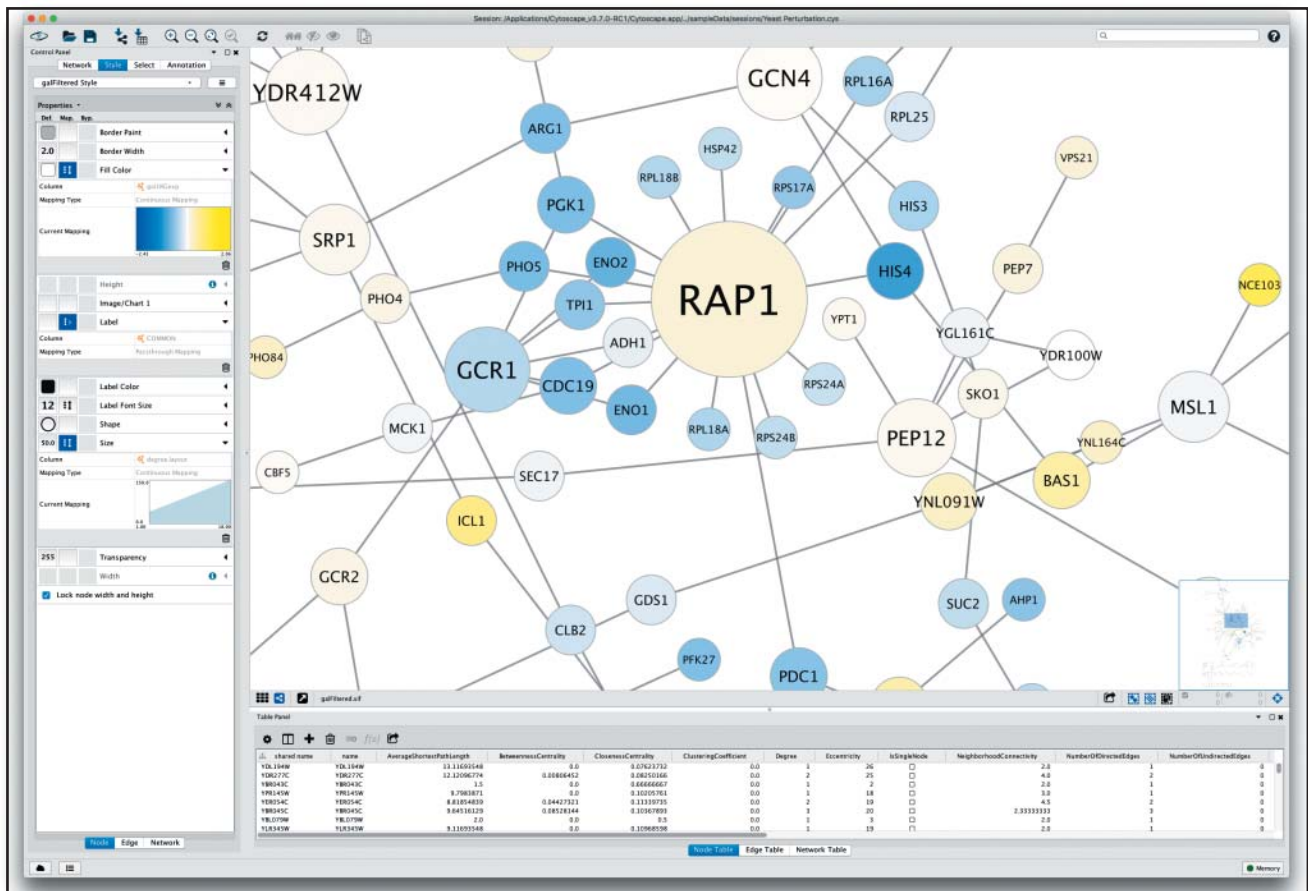


Figure 13.18 Zooming in on a network in Cytoscape shows part of a large connected network of protein and genetic interactions from budding yeast. This view is meant to emphasize the visual customization available in Cytoscape. Nodes represent genes, colored by gene expression values (similar to PathVisio) and sized by node degree. Importantly, this view in Cytoscape is highly customizable using Cytoscape’s visual mapper (left side panel).

Network Analysis

Many types of biological data can be integrated with biological networks for the purpose of gaining insight into mechanisms active in a particular biological context, such as disease. This section briefly describes a range of network analysis methods that have been developed, all of which are freely available as tools via Cytoscape or other systems.

Topological analysis concerns only studying the patterns of node and edge connections in a network. The most basic type of topological analysis is analyzing the distribution of the degree (the number of connections) across all nodes in the network. Biological network degree distributions tend to follow a power law, where a few nodes have a very high degree (called “hubs”) and many nodes have a low degree. It has been proposed that this organization underlies the robustness of biological systems (Barabasi and Oltvai 2004) and that hubs are the most important nodes in these systems. Many measures of node importance exist, often called centrality measures. For example, the “clustering coefficient” measures the density of edges in a node’s neighborhood. These measures can be computed using the Network Analyzer app that is available through Cytoscape (Assenov et al. 2008), as well as by other tools. Next, small patterns in networks, called network motifs, can be identified (Alon 2007). For instance, a feed-forward loop is composed of at least three nodes, where nodes are connected in series by directed edges (e.g. $A \rightarrow B \rightarrow C$) and where A is also connected to C by a directed edge, where direction indicates “regulation.” Many biological networks are enriched for particular motif types and simulations show that these motifs have specific biological properties, such as “delay” or “amplification.” Network motifs can be found using the NetMatchStar Cytoscape app (Rinnone et al. 2015). Next, larger patterns in networks are called modules (also known as systems or clusters), corresponding to nodes that are more connected to each other than they are to nodes outside the module. Modules in protein–protein interaction networks tend to be protein complexes (Bader and Hogue 2003) and modularity is a key principle of biological systems (Hartwell et al. 1999). Network modules can be deduced using the ClusterMaker2 Cytoscape app (Morris et al. 2011).

A second class of network analysis is differential analysis, where two or more networks are compared or aligned (Ideker and Krogan 2012). This analysis approach is useful in identifying regions that are conserved over evolutionary time and, thus, may be generally important. It can also identify regions that are different between conditions, such as regions that are disease specific, possibly aiding in the understanding of underlying disease mechanisms. The DyNet Cytoscape app is an example of this kind of network comparison tool (Goenawan et al. 2016).

A third class of network analysis is predictive, where networks are used to classify samples or predict disease outcome. For instance, classifying cancer samples by their mutation patterns relies on grouping samples having mutations in common. However, many cancer sample matches do not have any mutations in common, preventing any possible grouping. Cancer is thought of as a pathway disease, where cancer hallmark pathways need to be activated (tumor promoting) or deactivated (tumor suppressing); these effects can happen via many mutational mechanisms, not just repeatedly affecting the same genes (Hanahan and Weinberg 2011). Thus, even if two samples have different sets of mutations, the genes affected by the mutations may interact within the same modules and these relationships can be detected using gene functional interaction networks. A method called “network smoothing” has been proposed to “diffuse” information about mutations in each sample over a gene interaction network, increasing the similarity at the network level between samples. The sample-derived interaction networks can then be grouped more effectively than by mutations alone (Hofree et al. 2013).

A fourth class is network inference, where network edges are predicted from existing data. Two main types of network inference have been proposed: protein sequence and correlation based. Protein sequence-based methods use machine learning to identify patterns in protein sequences that are predictive of physical interactions (Schoenrock et al. 2014). Correlation-based methods identify correlations in a given set of data to define a network.

Examples include weighted correlation network analysis (WGCNA), which computes expression profile correlation scores between all gene pairs in a gene expression dataset (Langfelder and Horvath 2008). Correlation scores are mapped to weighted edges in a correlation network. The network is filtered to keep the strongest correlations and clustered to identify modules. Another example is ARACNE (Algorithm for the Reconstruction of Accurate Cellular Networks), which uses the mutual information measure and specialized filtering methods (the data processing inequality) to build a correlation network enriched for direct correlations among known regulators (e.g. transcription factors) and potential targets. The resulting network is predicted to correspond to gene regulatory relationships (Margolin et al. 2006). The Cytoscape CyNI toolkit app implements this and related methods (Guitart-Pla et al. 2015).

A final class is integrative analysis, where multiple layers of data are used to perform one of the above analysis types. The benefits of data integration are that errors from independent data sources are often reduced (increasing confidence), because each source is expected to generate error in different ways, and that coverage of a system can be increased, because each data source may have information about a different system aspect. Challenges include handling data matching to avoid errors, including matching gene identifiers or data types (e.g. continuous vs. discrete), and considering dataset bias (e.g. one of the data sources may interfere with the integration because it is biased or error prone). Data integration has been applied to predict protein interactions from multiple data types (Jain and Bader 2016), network module identification (Wang et al. 2014), and network modules affected by cancer mutations (Wu and Stein 2012). The last functionality is available in the ReactomeFIViz Cytoscape app. This app accepts a list of genes that is then queried against a functional interaction network created by

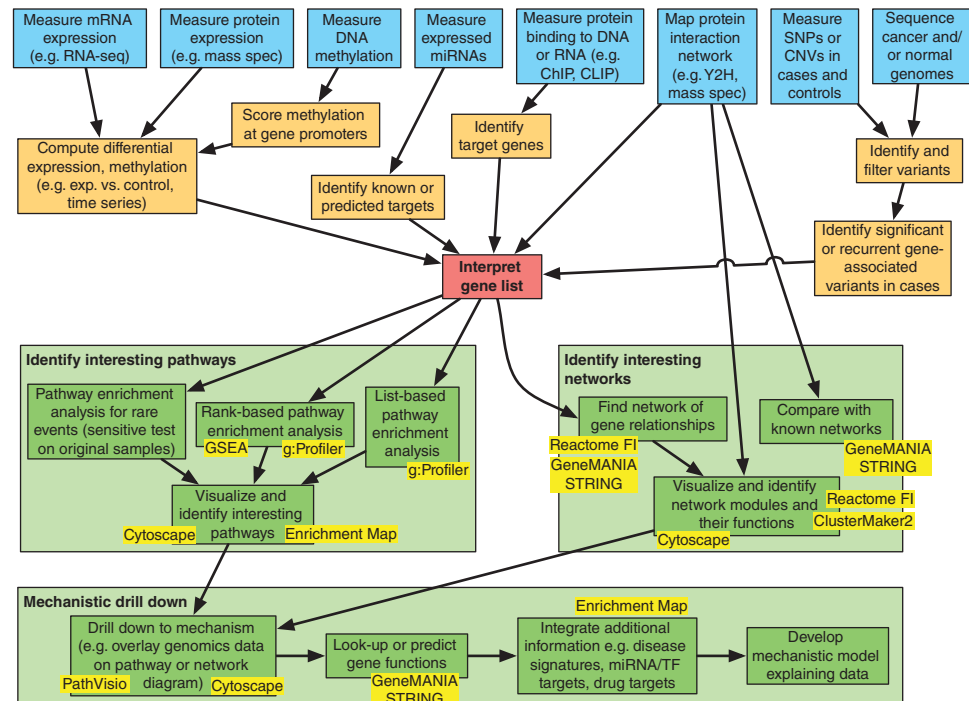


Figure 13.19 An overview of a pathway analysis workflow, summarizing multiple tools in this chapter. The top layer in blue depicts different genomics data types. The next layer in light orange shows data processing steps required to derive a gene list from the data. The gene list is represented by a red box. The green boxes describe data analysis and interpretation steps, with pathway enrichment analysis on the left and network analysis on the right. Both parallel approaches lead to focused analysis of pathways, network regions, and genes of interest (bottom green section). Yellow rectangles highlight tools discussed in this chapter. Arrows connect boxes to show paths through the overall workflow. ChIP, chromatin immunoprecipitation; CLIP, cross-linking immunoprecipitation; CNV, copy number variant; exp., experiment; GSEA, Gene Set Enrichment Analysis; mass spec., mass spectrometry; miRNA, mitochondrial RNA; SNP, single nucleotide polymorphism; TF, transcription factor.

integrating a range of data (including protein interactions); modules are then detected using a network clustering algorithm. Finally, each module is annotated using pathway enrichment analysis methods.

Summary

Given the wide range of pathway and network analysis methods available, it is difficult to select appropriate analysis methods that will work across any or all given data types. In terms of gene list interpretation, a good workflow involves identifying interesting pathways using pathway enrichment analysis methods. As pathway analysis focuses on known pathways, it does not include many genes from a typical genome, and network analysis should also be completed in parallel, using GeneMANIA and ReactomeFIViz within Cytoscape, to identify interesting network regions. Select interesting pathways and networks and participating genes can then be zoomed in on while manually considering all available data and literature to generate hypotheses to be experimentally tested (Figure 13.19).

Network and pathway information continues to rapidly grow, though it is typically represented as static information and is missing information about dynamics (e.g. a calcium wave or a feedback loop), detail (e.g. atomic protein structures), and context (e.g. cell type and developmental stage). Much work remains to develop representation and analysis methods that consider all available data about biological mechanisms in the cell to improve our ability to identify biological patterns and make testable predictions about biological systems. Many other topics about molecular interactions and pathways exist beyond what has been covered in this chapter. A sample of these are mathematical pathway modeling (Bower and Bolouri 2001), molecular docking of proteins with proteins and proteins with small molecules (Ofran and Rost 2003), and genetic interactions (Boone et al. 2007).

Acknowledgments

The author thanks Anton Enright for co-authoring the version of this chapter appearing in the previous edition of this book.

Internet Resources

BioCyc	biocyc.org
BioGRID	thebiogrid.org
BioPAX	www.biopax.org
Cytoscape	www.cytoscape.org
GeneMANIA	genemania.org
g:Profiler	biit.cs.ut.ee/gprofiler
Human Protein Reference Database (HPRD)	www.hprd.org
IntAct	www.ebi.ac.uk/intact
The Kyoto Encyclopedia of Genes and Genomes (KEGG)	www.genome.jp/kegg/kegg2.html
Pathguide	www.pathguide.org
Proteomics Standards Initiative–Molecular Interactions (PSI-MI)	www.psidev.info/groups/molecular-interactions
Reactome	reactome.org
Systems Biology Graphical Notation (SBGN)	sbgn.github.io/sbgn
Systems Biology Markup Language (SBML)	sbml.org
STRING	string-db.org

Further Reading

- Barabasi, A.L. and Oltvai, Z.N. (2004). Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.* 5 (2): 101–113. This review explains the concept of network analysis to understand the cell's functional organization.
- Ideker, T., Galitski, T., and Hood, L. (2001). A new approach to decoding life: systems biology. *Annu. Rev. Genomics Hum. Genet.* 2: 343–372. This review helps define the field of systems biology. Pathway and network information is required for input to systems biology analysis methods and is an output of systems biology experimental methods.
- Merico, D., Gfeller, D., and Bader, G.D. (2009). How to visually interpret biological data using networks. *Nat. Biotechnol.* 27 (10): 921–924. This short primer explains how to visually interpret networks.
- Reimand, J., Isserlin, R., Voisin, V. et al. (2019). Pathway enrichment analysis and visualization of omics data using g:Profiler, GSEA, Cytoscape and EnrichmentMap. *Nat. Protoc.* 14 (2): 482–517. This protocol describes how to perform major types of pathway enrichment analysis.

References

- Akiva, E., Brown, S., Almonacid, D.E. et al. (2014). The structure-function linkage database. *Nucleic Acids Res.* 42 (Database issue): D521–D530.
- Alon, U. (2007). Network motifs: theory and experimental approaches. *Nat. Rev. Genet.* 8 (6): 450–461.
- Ashburner, M., Ball, C.A., Blake, J.A. et al. (2000). Gene Ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.* 25 (1): 25–29.
- Assenov, Y., Ramirez, F., Schelhorn, S.E. et al. (2008). Computing topological parameters of biological networks. *Bioinformatics* 24 (2): 282–284.
- Bader, G.D. and Hogue, C.W. (2002). Analyzing yeast protein-protein interaction data obtained from different sources. *Nat. Biotechnol.* 20 (10): 991–997.
- Bader, G.D. and Hogue, C.W. (2003). An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinf.* 4 (2).
- Bader, G.D., Cary, M.P., and Sander, C. (2006). Pathguide: a pathway resource list. *Nucleic Acids Res.* 34 (Database issue): D504–D506.
- Barabasi, A.L. and Oltvai, Z.N. (2004). Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.* 5 (2): 101–113.
- Bindea, G., Mlecnik, B., Hackl, H. et al. (2009). ClueGO: a Cytoscape plug-in to decipher functionally grouped gene ontology and pathway annotation networks. *Bioinformatics* 25 (8): 1091–1093.
- Bollobás, B. (1998). *Modern Graph Theory*. New York, NY: Springer.
- Boone, C., Bussey, H., and Andrews, B.J. (2007). Exploring genetic interactions and networks with yeast. *Nat. Rev. Genet.* 8 (6): 437–449.
- Bower, J.M. and Bolouri, H. (2001). *Computational Modeling of Genetic and Biochemical Networks*. Cambridge, MA: MIT Press.
- Braun, P., Tasan, M., Dreze, M. et al. (2009). An experimentally derived confidence score for binary protein-protein interactions. *Nat. Methods* 6 (1): 91–97.
- Brown, K.R. and Jurisica, I. (2007). Unequal evolutionary conservation of human protein interactions in interologous networks. *Genome Biol.* 8 (5): R95.
- Caspi, R., Billington, R., Fulcher, C.A. et al. (2018). The MetaCyc database of metabolic pathways and enzymes. *Nucleic Acids Res.* 46 (D1): D633–D639.
- Cerami, E.G., Gross, B.E., Demir, E. et al. (2011). Pathway commons, a web resource for biological pathway data. *Nucleic Acids Res.* 39 (Database issue): D685–D690.
- Chatr-Aryamontri, A., Oughtred, R., Boucher, L. et al. (2017). The BioGRID interaction database: 2017 update. *Nucleic Acids Res.* 45 (Database issue): D369–D379.

- Cormen, T.H. (2001). *Introduction to Algorithms*. Cambridge, MA: MIT Press.
- Dandekar, T., Snel, B., Huynen, M., and Bork, P. (1998). Conservation of gene order: a fingerprint of proteins that physically interact. *Trends Biochem. Sci.* 23 (9): 324–328.
- Demchak, B., Otasek, D., Pico, A.R. et al. (2018). The Cytoscape Automation app article collection. *F1000Research* 7: 800.
- Demir, E., Cary, M.P., Paley, S. et al. (2010). The BioPAX community standard for pathway data sharing. *Nat. Biotechnol.* 28 (9): 935–942.
- Demir, E., Babur, O., Rodchenkov, I. et al. (2013). Using biological pathway data with paxtools. *PLoS Comput. Biol.* 9 (9): e1003194.
- Eeckman, F.H. and Durbin, R. (1995). ACeDB and macace. *Methods Cell Biol.* 48: 583–605.
- Enright, A.J., Iliopoulos, I., Kyripides, N.C., and Ouzounis, C.A. (1999). Protein interaction maps for complete genomes based on gene fusion events. *Nature* 402 (6757): 86–90.
- Enright, A.J., Van Dongen, S., and Ouzounis, C.A. (2002). An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res.* 30 (7): 1575–1584.
- Evsikov, A.V., Dolan, M.E., Genrich, M.P. et al. (2009). MouseCyc: a curated biochemical pathways database for the laboratory mouse. *Genome Biol.* 10 (8): R84.
- Fabregat, A., Jupe, S., Matthews, L. et al. (2018). The reactome pathway knowledgebase. *Nucleic Acids Res.* 46 (D1): D649–D655.
- Franz, M., Rodriguez, H., Lopes, C. et al. (2018). GeneMANIA update 2018. *Nucleic Acids Res.* 46 (Web Server issue): W60–W64.
- Gavin, A.C., Bosche, M., Krause, R. et al. (2002). Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature* 415 (6868): 141–147.
- Ge, H., Liu, Z., Church, G.M., and Vidal, M. (2001). Correlation between transcriptome and interactome mapping data from *Saccharomyces cerevisiae*. *Nat. Genet.* 29 (4): 482–486.
- Goenawan, I.H., Bryan, K., and Lynn, D.J. (2016). DyNet: visualization and analysis of dynamic molecular interaction networks. *Bioinformatics* 32 (17): 2713–2715.
- 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 (17): 3513–3519.
- Guitart-Pla, O., Kustagi, M., Rugheimer, F. et al. (2015). The Cyni framework for network inference in Cytoscape. *Bioinformatics* 31 (9): 1499–1501.
- Hanahan, D. and Weinberg, R.A. (2011). Hallmarks of cancer: the next generation. *Cell* 144 (5): 646–674.
- Hartmann, A. and Jozefowicz, A.M. (2018). VANTED: a tool for integrative visualization and analysis of -omics data. *Methods Mol. Biol.* 1696: 261–278.
- Hartwell, L.H., Hopfield, J.J., Leibler, S., and Murray, A.W. (1999). From molecular to modular cell biology. *Nature* 402 (6761 Suppl): C47–C52.
- Helmy, M., Crits-Christoph, A., and Bader, G.D. (2016). Ten simple rules for developing public biological databases. *PLoS Comput. Biol.* 12 (11): e1005128.
- Hermjakob, H., Montecchi-Palazzi, L., Bader, G. et al. (2004). The HUPO PSI's molecular interaction format – a community standard for the representation of protein interaction data. *Nat. Biotechnol.* 22 (2): 177–183.
- Ho, Y., Gruhler, A., Heilbut, A. et al. (2002). Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry. *Nature* 415 (6868): 180–183.
- Hochberg, Y. and Benjamini, Y. (1990). More powerful procedures for multiple significance testing. *Stat. Med.* 9 (7): 811–818.
- Hofree, M., Shen, J.P., Carter, H. et al. (2013). Network-based stratification of tumor mutations. *Nat. Methods* 10 (11): 1108–1115.
- Hucka, M., Finney, A., Sauro, H.M. et al. (2003). The Systems Biology Markup Language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* 19 (4): 524–531.
- Ideker, T. and Krogan, N.J. (2012). Differential network biology. *Mol. Syst. Biol.* 8: 565.

- Ito, T., Tashiro, K., Muta, S. et al. (2000). Toward a protein-protein interaction map of the budding yeast: a comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins. *Proc. Natl. Acad. Sci. U.S.A.* 97 (3): 1143–1147.
- Jain, S. and Bader, G.D. (2016). Predicting physiologically relevant SH3 domain mediated protein-protein interactions in yeast. *Bioinformatics* 32 (12): 1865–1872.
- Jansen, R., Greenbaum, D., and Gerstein, M. (2002). Relating whole-genome expression data with protein-protein interactions. *Genome Res.* 12 (1): 37–46.
- Jansen, R., Yu, H., Greenbaum, D. et al. (2003). A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science* 302 (5644): 449–453.
- Kanehisa, M., Goto, S., Kawashima, S., and Nakaya, A. (2002). The KEGG databases at GenomeNet. *Nucleic Acids Res.* 30 (1): 42–46.
- Karp, P.D. and Riley, M. (1993). Representations of metabolic knowledge. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 1: 207–215.
- Karp, P.D., Riley, M., Saier, M. et al. (2002). The EcoCyc database. *Nucleic Acids Res.* 30 (1): 56–58.
- Karp, P.D., Latendresse, M., and Caspi, R. (2011). The pathway tools pathway prediction algorithm. *Stand. Genomic Sci.* 5 (3): 424–429.
- Keseler, I.M., Mackie, A., Santos-Zavaleta, A. et al. (2017). The EcoCyc database: reflecting new knowledge about *Escherichia coli* K-12. *Nucleic Acids Res.* 45 (Database issue): D543–D550.
- Khatri, P., Sirota, M., and Butte, A.J. (2012). Ten years of pathway analysis: current approaches and outstanding challenges. *PLoS Comput. Biol.* 8 (2): e1002375.
- Kim, M.S., Pinto, S.M., Getnet, D. et al. (2014). A draft map of the human proteome. *Nature* 509 (7502): 575–581.
- Kotlyar, M., Pastrello, C., Pivetta, F. et al. (2015). In silico prediction of physical protein interactions and characterization of interactome orphans. *Nat. Methods* 12 (1): 79–84.
- Kutmon, M., van Iersel, M.P., Bohler, A. et al. (2015). PathVisio 3: an extendable pathway analysis toolbox. *PLoS Comput. Biol.* 11 (2): e1004085.
- Langfelder, P. and Horvath, S. (2008). WGCNA: an R package for weighted correlation network analysis. *BMC Bioinf.* 9: 559.
- Le Novere, N., Bornstein, B., Broicher, A. et al. (2006). BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Res.* 34 (Database issue): D689–D691.
- Le Novere, N., Hucka, M., Mi, H. et al. (2009). The systems biology graphical notation. *Nat. Biotechnol.* 27 (8): 735–741.
- Li, P., Li, J., Wang, L., and Di, L.J. (2017). Proximity labeling of interacting proteins: application of BioID as a discovery tool. *Proteomics* 17 (20): 1–10.
- Lotia, S., Montojo, J., Dong, Y. et al. (2013). Cytoscape app store. *Bioinformatics* 29 (10): 1350–1351.
- Luck, K., Sheynkman, G.M., Zhang, I., and Vidal, M. (2017). Proteome-scale human interactomics. *Trends Biochem. Sci.* 42 (5): 342–354.
- Mack, S.C., Witt, H., Piro, R.M. et al. (2014). Epigenomic alterations define lethal CIMP-positive ependymomas of infancy. *Nature* 506 (7489): 445–450.
- Marcotte, E.M., Pellegrini, M., Ng, H.L. et al. (1999). Detecting protein function and protein-protein interactions from genome sequences. *Science* 285 (5428): 751–753.
- Margolin, A.A., Nemenman, I., Basso, K. et al. (2006). ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinf.* 7 (Suppl 1): S7.
- Matthews, L.R., Vaglio, P., Reboul, J. et al. (2001). Identification of potential interaction networks using sequence-based searches for conserved protein-protein interactions or “interologs”. *Genome Res.* 11 (12): 2120–2126.
- Mayer, G., Jones, A.R., Binz, P.A. et al. (2014). Controlled vocabularies and ontologies in proteomics: overview, principles and practice. *Biochim. Biophys. Acta* 1844 (1 Pt A): 98–107.
- Mehlhorn, K. and Nèaher, S. (1999). *Leda: A Platform for Combinatorial and Geometric Computing*. New York, NY: Cambridge University Press.

- Meldal, B.H., Forner-Martinez, O., Costanzo, M.C. et al. (2015). The complex portal – an encyclopaedia of macromolecular complexes. *Nucleic Acids Res.* 43 (Database issue): D479–D484.
- Merico, D., Gfeller, D., and Bader, G.D. (2009). How to visually interpret biological data using networks. *Nat. Biotechnol.* 27 (10): 921–924.
- Merico, D., Isserlin, R., Stueker, O. et al. (2010). Enrichment map: a network-based method for gene-set enrichment visualization and interpretation. *PLoS One* 5 (11): e13984.
- Morris, J.H., Apeltsin, L., Newman, A.M. et al. (2011). clusterMaker: a multi-algorithm clustering plugin for Cytoscape. *BMC Bioinf.* 12 (1): 436.
- Ofran, Y. and Rost, B. (2003). Analysing six types of protein-protein interfaces. *J. Mol. Biol.* 325 (2): 377–387.
- Ouzounis, C. and Kyriades, N. (1996). The emergence of major cellular processes in evolution. *FEBS Lett.* 390 (2): 119–123.
- Overbeek, R., Fonstein, M., D'Souza, M. et al. (1999). The use of gene clusters to infer functional coupling. *Proc. Natl. Acad. Sci. U.S.A.* 96 (6): 2896–2901.
- Pellegrini, M., Marcotte, E.M., Thompson, M.J. et al. (1999). Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc. Natl. Acad. Sci. U.S.A.* 96 (8): 4285–4288.
- Phizicky, E.M. and Fields, S. (1995). Protein-protein interactions: methods for detection and analysis. *Microbiol. Rev.* 59 (1): 94–123.
- Pinto, D., Pagnamenta, A.T., Klei, L. et al. (2010). Functional impact of global rare copy number variation in autism spectrum disorders. *Nature* 466: 368–372.
- Pouliot, Y. and Karp, P.D. (2007). A survey of orphan enzyme activities. *BMC Bioinf.* 8: 244.
- Razick, S., Magklaras, G., and Donaldson, I.M. (2008). iRefIndex: a consolidated protein interaction database with provenance. *BMC Bioinf.* 9: 405.
- Reimand, J., Arak, T., Adler, P. et al. (2016). g:Profiler—a web server for functional interpretation of gene lists (2016 update). *Nucleic Acids Res.* 44 (W1): W83–W89.
- Reimand, J., Isserlin, R., Voisin, V. et al. (2019). Pathway enrichment analysis and visualization of omics data using g:Profiler, GSEA, Cytoscape and EnrichmentMap. *Nat. Protoc.* 14 (2): 482–517.
- Rinnone, F., Micale, G., Bonnici, V. et al. (2015). NetMatchStar: an enhanced Cytoscape network querying app. *F1000Research* 4: 479.
- Rivera, M.C., Jain, R., Moore, J.E., and Lake, J.A. (1998). Genomic evidence for two functionally distinct gene classes. *Proc. Natl. Acad. Sci. U.S.A.* 95 (11): 6239–6244.
- Rodchenkov, I., Demir, E., Sander, C., and Bader, G.D. (2013). The BioPAX validator. *Bioinformatics* 29 (20): 2659–2660.
- Sari, M., Bahceci, I., Dogrusoz, U. et al. (2015). SBGNViz: a tool for visualization and complexity management of SBGN process description maps. *PLoS One* 10 (6): e0128985.
- Schoenrock, A., Samanfar, B., Pitre, S. et al. (2014). Efficient prediction of human protein-protein interactions at a global scale. *BMC Bioinf.* 15: 383.
- Schwikowski, B., Uetz, P., and Fields, S. (2000). A network of protein-protein interactions in yeast. *Nat. Biotechnol.* 18 (12): 1257–1261.
- Shannon, P., Markiel, A., Ozier, O. et al. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.* 13 (11): 2498–2504.
- Snel, B., Bork, P., and Huynen, M.A. (2002). Genomes in flux: the evolution of archaeal and proteobacterial gene content. *Genome Res.* 12 (1): 17–25.
- Subramanian, A., Tamayo, P., Mootha, V.K. et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. U.S.A.* 102 (43): 15545–15550.
- Szklarczyk, D., Franceschini, A., Wyder, S. et al. (2015). STRING v10: protein-protein interaction networks, integrated over the tree of life. *Nucleic Acids Res.* 43 (Database issue): D447–D452.
- Tamames, J., Casari, G., Ouzounis, C., and Valencia, A. (1997). Conserved clusters of functionally related genes in two bacterial genomes. *J. Mol. Evol.* 44 (1): 66–73.

- Tavazoie, S., Hughes, J.D., Campbell, M.J. et al. (1999). Systematic determination of genetic network architecture. *Nat. Genet.* 22 (3): 281–285.
- Tien, A.C., Lin, M.H., Su, L.J. et al. (2004). Identification of the substrates and interaction proteins of aurora kinases from a protein-protein interaction model. *Mol. Cell. Proteomics* 3 (1): 93–104.
- Tong, A.H., Evangelista, M., Parsons, A.B. et al. (2001). Systematic genetic analysis with ordered arrays of yeast deletion mutants. *Science* 294 (5550): 2364–2368.
- Toure, V., Le Novere, N., Waltemath, D., and Wolkenhauer, O. (2018). Quick tips for creating effective and impactful biological pathways using the systems biology graphical notation. *PLoS Comput. Biol.* 14 (2): e1005740.
- Turei, D., Korcsmaros, T., and Saez-Rodriguez, J. (2016). OmniPath: guidelines and gateway for literature-curated signaling pathway resources. *Nat. Methods* 13 (12): 966–967.
- van Iersel, M.P., Villeger, A.C., Czauderna, T. et al. (2012). Software support for SBGN maps: SBGN-ML and LibSBGN. *Bioinformatics* 28 (15): 2016–2021.
- Voet, D. and Voet, J.G. (2004). *Biochemistry*. New York, NY: Wiley.
- Walhout, A.J., Boulton, S.J., and Vidal, M. (2000). Yeast two-hybrid systems and protein interaction mapping projects for yeast and worm. *Yeast* 17 (2): 88–94.
- Wang, B., Mezlini, A.M., Demir, F. et al. (2014). Similarity network fusion for aggregating data types on a genomic scale. *Nat. Methods* 11 (3): 333–337.
- Wu, G. and Stein, L. (2012). A network module-based method for identifying cancer prognostic signatures. *Genome Biol.* 13 (12): R112.
- Zeeberg, B.R., Riss, J., Kane, D.W. et al. (2004). Mistaken identifiers: gene name errors can be introduced inadvertently when using Excel in bioinformatics. *BMC Bioinf.* 5: 80.

14

Metabolomics

David S. Wishart

Introduction

Most of this textbook describes the computational tools and databases needed to facilitate research into genomics, transcriptomics, and proteomics. In other words, the molecular entities of interest are primarily large molecules or mega-polymers such as proteins, RNA, and DNA. What about the study of small molecules such as amino acids, nucleotides, and lipids? Over the past decade, an increasing number of bioinformaticians have been turning their attention toward these small molecules through an emerging field of science called metabolomics. Metabolomics is a branch of “omics” science that is focused on the comprehensive characterization of the small molecule metabolites in the metabolome. The metabolome is defined as the complete collection of all small molecules (with a molecular weight <1500 Da) found in a cell, a biofluid, an organ, or an organism (Wishart 2005). These small molecules include endogenous metabolites such as short peptides, amino acids, nucleic acids, carbohydrates, lipids, organic acids, vitamins, and minerals. They also include exogenous chemicals or xenobiotics such as food additives, plant phytochemicals, drugs, cosmetic chemicals, dyes, detergents, pollutants, or just about any other small molecule chemical that an organism can consume or to which it can be exposed.

Small molecules are essential to life. They are the bricks and mortar of cells, serving as the building blocks for all the macromolecules (the proteins, RNA, and DNA) needed for basic cellular functions. They also provide the fuel for cellular processes, the fences to maintain cellular integrity, the buffers to help cells tolerate environmental stressors, and the messengers for many intracellular and intercellular signaling events. As many small molecule metabolites are “encoded” by specific genes and because they play such a vital role in nearly all cellular processes, metabolites are sometimes called the “canaries of the genome.” Just as canaries served as sensitive indicators of toxic gases or other problems in coal mines, small molecule metabolites can serve as exquisitely sensitive indicators of problems in the genome. Indeed, a single base change in a gene can lead to a 10 000-fold change in the concentrations of certain metabolites (Wishart et al. 2007). This remarkable sensitivity is the basis for newborn screening, in which metabolite tests have been used to detect genetic defects (such as phenylketonuria) for many decades (Levy 2010). Metabolite levels are not only very sensitive to what goes on in the genome, they are also very sensitive to what goes on in the environment. Indeed, metabolite levels are heavily influenced by nutrition, activity, exposure to noxious chemicals, the time of day, or even the outside temperature (Bassini and Cameron 2014; Brown 2016).

As metabolites are the end products of complex interactions happening inside the cell (the genome) and events happening outside the cell (the environment), metabolomic approaches permit the comprehensive assessment of interactions between genes and the environment. Since the end product of an organism’s genotype and its environmental interactions (i.e. genotype \times environment) is defined as its phenotype, metabolomics offers an ideal route for scientists to measure, in real time, the phenotype or physiological state of an organism

(Fiehn 2002). This represents an important advantage of metabolomics over genomics. While the genome can tell you what *might* happen, the metabolome actually tells you what is happening.

Continuing advances in both analytical chemistry and computational data analysis techniques are now making metabolomics far more accessible to a much wider range of scientific disciplines. Indeed, metabolomics is now routinely used in disease screening, in biomedical research, in drug discovery, in food and nutritional analysis, in veterinary studies, in crop assessment, in biomaterial production, and in environmental monitoring (Holmes et al. 2008; Viant 2008; Kim et al. 2016; Wishart 2016). These metabolomic studies have led to some remarkable discoveries, such as the identification of microbially derived trimethylamine oxide (TMAO) as being one of the key drivers behind atherosclerosis (Wang et al. 2011a) and the determination that high serum levels of branched chain amino acids can predict who will develop type 2 diabetes 10–15 years before the disease actually develops (Wang et al. 2011b). As a result, metabolomics has experienced tremendous growth, with just two metabolomic papers published in 1999 to more than 3100 in 2016.

A diagram illustrating the typical workflow for a metabolomic experiment is shown in Figure 14.1. Initially, a biological sample (such as a tissue, an organ, a plant, cells from a cell culture, or even an environmental sample) is collected. Then, it is metabolically quenched using liquid nitrogen or other rapid freezing techniques, after which it is extracted or homogenized to produce a liquid mixture containing hundreds of metabolites. In most cases, it is far easier to collect a biofluid, such as blood, urine, tree sap, or a cell growth medium, as this avoids the tissue powdering/extraction process. Once an appropriate

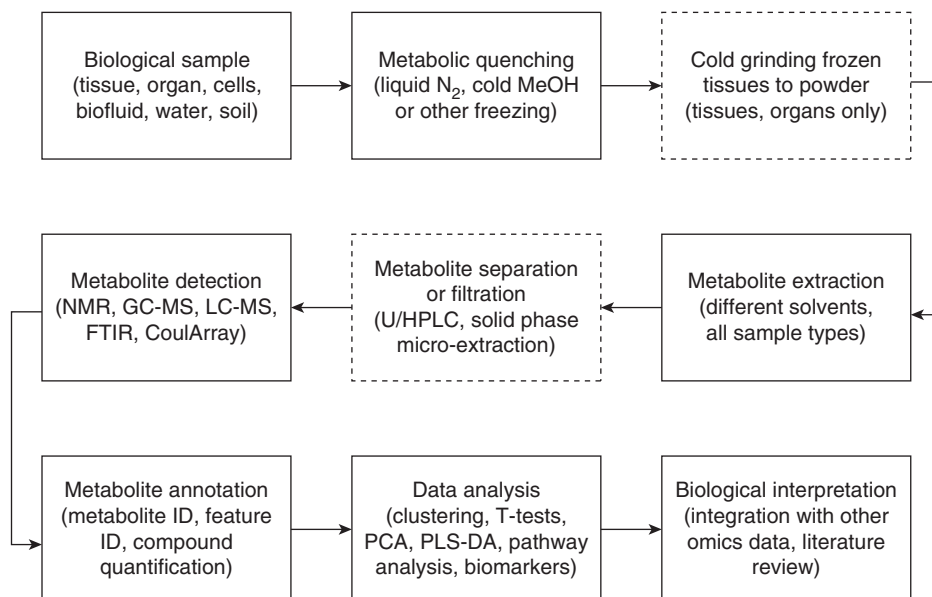


Figure 14.1 A diagram illustrating the typical workflow for a metabolomic experiment. Boxes with solid lines represent steps required in all metabolomic experiments. Boxes with dashed lines represent steps that are sample dependent. Samples (e.g. tissues, organs, cells, or biofluids) are initially collected and then rapidly “quenched” to stop any metabolic reactions. If tissues are used, the samples must be ground to a fine powder (while frozen). Samples are extracted to obtain the metabolites and then separated or filtered (not all samples need to be separated/filtered). After this step, samples can be analyzed via nuclear magnetic resonance (NMR), gas chromatography–mass spectrometry (GC-MS), liquid chromatography–mass spectrometry (LC-MS) or other analytical chemistry techniques. The resulting spectra are then processed and the metabolites annotated. The annotated data are further analyzed using various statistical and visualization techniques. Finally, the metabolomic data are integrated with other kinds of information for further biological interpretation. FTIR, Fourier transform infrared; MeOH, methanol; PCA, principal component analysis; PLS-DA, partial least squares discriminant analysis; U/HPLC, ultra-high-performance liquid chromatography.

metabolite extract or biofluid has been obtained, it then needs to be run through one or more analytical chemistry platforms. These analytical platforms may be mass spectrometry (MS) instruments equipped with liquid chromatography (LC) or gas chromatography (GC) systems, or they may be nuclear magnetic resonance (NMR) instruments. Other kinds of analytical tools may occasionally be used, such as Fourier transform infrared spectrometers (FTIR) or coulometric (electrochemical) array systems. These kinds of analytical tools are capable of separating, detecting, and characterizing hundreds (or even thousands) of chemicals in complex chemical mixtures. In almost all cases, NMR, GC-MS, or LC-MS instruments produce spectra or chromatograms consisting of many hundreds to thousands of peaks. The primary bioinformatic challenge in metabolomics, therefore, is having the appropriate software tools to determine which peaks in these spectra match which chemical compounds (i.e. metabolite annotation in Figure 14.1). The secondary bioinformatic challenge is having the appropriate software to determine which compounds or spectral peaks have changed significantly and why (i.e. data analysis and biological interpretation in Figure 14.1).

This chapter is intended to provide an overview of the bioinformatic tools and databases needed to perform metabolomic analyses. It is organized into six sections: a short introduction to metabolomics, a description of the different data formats for metabolomics, a brief review of the major metabolomic databases, a description of common bioinformatic tools for metabolite identification or annotation, a summary of selected bioinformatic tools for multivariate data analysis and visualization, and a description of several bioinformatic tools for metabolite and/or biological interpretation. Readers wishing to learn more about the technologies and analytical tools used in metabolomics are encouraged to read comprehensive reviews covering these subjects (Dunn et al. 2005; Wishart 2008; Naz et al. 2014).

Data Formats

Metabolomic data are fundamentally different than genomic or proteomic data. As seen in Chapter 1, genomic or proteomic data typically consist of gene or protein sequences in FASTA format (for sequence files) or FASTQ format (for sequence reads). On the other hand, metabolomic data generally consist of chemical names, chemical identifiers, chemical structures, and their corresponding MS or NMR spectra. Therefore, most of the data formats and formatting rules for metabolomic data tend to fall under the jurisdiction of chemistry (rather than molecular biology) and cheminformatics (rather than bioinformatics). These chemical data standards are governed by rules and recommendations established by the International Union of Pure and Applied Chemistry (IUPAC).

In genomics or proteomics, if a new gene or protein is identified, it is often named according to its function (e.g. “alcohol dehydrogenase”). Alternately, if no function is immediately apparent, it is possible to give the gene/protein a completely whimsical name such as “Sonic Hedgehog” or “Reaper.” On the other hand, if a new chemical is identified, its official name is formally defined by its structure, using strict IUPAC nomenclature rules. These nomenclature rules are sufficiently well designed that almost any chemical can be automatically named (via computer programs) using only its structure as input. More recently, software has been developed that supports the reverse process (i.e. name to structure). Several commercial software packages, as well as a number of open access software tools and web servers such as Openmolecules.org and the OPSIN web server, can perform these name-to-structure and structure-to-name operations. While IUPAC naming conventions have been adopted universally, there is still widespread use of common names, brand names, and synonymous or trivial names for many compounds, especially in metabolomics. Given the ambiguity of common or trivial chemical names, many metabolomic researchers have turned to using chemical structures or standardized chemical identifiers to help eliminate this ambiguity. Some of these are outlined below.


```

L-alanine
Mrv1722704261716482D

6 5 0 0 1 0      999 V2000
 0.6740 -0.6740  0.0000 N  0 0 0 0 0 0 0 0 0 0 0 0
 0.6740  0.1510  0.0000 C  0 0 1 0 0 0 0 0 0 0 0 0
-0.0405  0.5635  0.0000 C  0 0 0 0 0 0 0 0 0 0 0 0
 1.3885  0.5635  0.0000 C  0 0 0 0 0 0 0 0 0 0 0 0
 1.3885  1.3885  0.0000 O  0 0 0 0 0 0 0 0 0 0 0 0
 2.1030  0.1510  0.0000 O  0 0 0 0 0 0 0 0 0 0 0 0
2 1 1 1 0 0 0
2 3 1 0 0 0 0
2 4 1 0 0 0 0
4 5 1 0 0 0 0
4 6 2 0 0 0 0
M END

```

Figure 14.3 An example of a MOL file for a two-dimensional representation of L-alanine. The first few rows are identifiers. The x,y,z coordinates are given in the first three columns with the alanine nitrogen atom having the coordinates 0.6740, -0.6740, 0.0000. The connection table (which of the six atoms are connected to each other) is given below the coordinate list. For instance, atom 2 (the alpha carbon) is connected to atom 1 (the nitrogen) by a single bond, which gives a connection list of 2 1 1 (in the first line of the connection table). Likewise, atom 4 (the carbonyl carbon) is connected to atom 6 (an oxygen atom) by a double bond, giving a connection list of 4 6 2 in the last line of the connection table.

commonly used 2D structure formats are the Structure Data Format (SDF) and Molfile (MOL) file formats (Dalby et al. 1992). An example of a 2D MOL file format for L-alanine is shown in Figure 14.3. The 3D structures of small molecules can also be represented using SDF and MOL file formats. In many respects, the SDF and MOL formats are the equivalent to the Protein Data Bank (PDB) format (Westbrook and Fitzgerald 2003) for representing protein, DNA, and RNA structures. Interestingly, the PDB format is also widely used to represent the 3D structures of small molecules. Furthermore, it is often possible to convert PDB formatted files into SDF or MOL formatted files using freely available data exchange tools such as Open Babel (O’Boyle et al. 2011).

Spectral Representation and Exchange Formats

In addition to having well-defined names, text representations and 2D (or 3D) structures, most small molecules need to be associated with specific “referential” NMR or MS spectra. These reference spectra provide not only experimental evidence for a compound’s existence but also a unique and often easily interpreted signature that clearly identifies that compound within the more complex spectrum of a biosample containing many compounds. The importance of spectral data in the field of metabolomics cannot be underestimated. Indeed, for most metabolomic experiments, the metabolites of interest must ultimately be identified via spectral matching using reference spectral libraries. These libraries contain thousands of carefully collected spectra of single, highly purified compounds. However, for spectral matching algorithms to work, the spectral library data format needs to be compatible with the query spectral format. Fortunately, there are now a number of common data exchange formats for storing, querying, and sharing NMR and MS spectral data.

Historically, the “official” format for exchanging small molecule NMR and MS spectral data was called JCAMP-DX, which stands for Joint Committee on Atomic and Molecular Physical Data eXtension. This data format was developed through the Joint Committee on Atomic and Molecular Physical Data in the 1980s (McDonald and Wilks 1988). However, JCAMP-DX is now quite outdated and is being superseded by a variety of more modern eXtensible Markup Language (XML) formats. These include the Chemical Markup Language or CML, which is somewhat generic (Kuhn et al. 2007); mzML (Deutsch 2017), which is used for handling mass spectral data; and nmrML (Schober et al. 2018), which is used for handling NMR spectral data. An example of a portion of an nmrML data file for L-alanine is shown in Figure 14.4.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<nmrML xmlns="http://nmrml.org/schema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://nmrml.org/schema http://nmrml.org/schema/v1.0.rc1/nmrML.xsd"
  version="v1.0.rc1" >
  <cvList>
    <cv id="NMRCV" fullName="Nuclear Magnetic Resonance CV" version="1.1.0" URI="http://nmrml.org/cv/v1.1.0/nmrCV.owl"/>
    <cv id="UO" fullName="Unit Ontology" version="3.2.0" URI="http://purl.obolibrary.org/obo"/>
    <cv id="CHEBI" fullName="Chemical Entities of Biological Interest Ontology" version="105" URI="http://
purl.obolibrary.org/obo"/>
    <cv id="NCIThesaurus" fullName="NCI Thesaurus" version="" URI="http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#"/>
  </cvList>
  <fileDescription>
    <fileContent>
      <cvParam cvRef="NMRCV" accession="NMR:1400165" name="1D NMR acquisition parameter set"/>
    </fileContent>
  </fileDescription>
  <contactList>
    <contact id="ID00001" fullName="undefined" email="undefined"/>
  </contactList>
  <sourceFileList>
    <sourceFile id="ID00002" name="fid" location="file:/apps/nmrml/project/releases/20161223213945/public/system/bayesil/
0f3673601af7ac591e8cb353f1b7f9b7b2b5f679/spectrum.fid/fid" sha1="747013506b54ca432ed1deeb6d015aa84361ec9e">
      <cvParam cvRef="NMRCV" accession="NMR:1400297" name="Varian VNMR format"/>
      <cvParam cvRef="NMRCV" accession="NMR:1400119" name="FID file"/>
    </sourceFile>
    <sourceFile id="ID00003" name="procpa" location="file:/apps/nmrml/project/releases/20161223213945/public/system/
bayesil/0f3673601af7ac591e8cb353f1b7f9b7b2b5f679/spectrum.fid/procpa">
      <cvParam cvRef="NMRCV" accession="NMR:1400297" name="Varian VNMR format"/>
      <cvParam cvRef="NMRCV" accession="NMR:1002006" name="acquisition parameter file"/>
    </sourceFile>
  </sourceFileList>
  <softwareList>
    <software id="ID00004" version="1.1" cvRef="NMRCV" accession="NMR:1000277" name="VnmrJ software"/><software id="ID00007"
cvRef="NMRCV" accession="NMR:1000183" name="nmrML Assign"/><software id="ID00006" cvRef="NMRCV" accession="NMR:1000183"
name="Bayesil software"/>
  </softwareList>
  <instrumentConfigurationList>
    <instrumentConfiguration id="ID00005">
      <cvParam cvRef="NMRCV" accession="NMR:1400234" name="Varian NMR instrument"/>
      <userParam name="ProbeHead" value="cold"/>
      <softwareRef ref="ID00004"/>
    </instrumentConfiguration>
  </instrumentConfigurationList>
  <acquisition>

```

Figure 14.4 An example of an nmrML data file for L-alanine. The actual file is many hundreds of lines long and includes a digital (byte format) representation of the actual nuclear magnetic resonance (NMR) spectrum of L-alanine. The value of the nmrML format lies in the header information, which provides rich data about how the NMR spectrum was acquired and processed.

These new(er) spectral data formats allow the capture of much more metadata (meaning data about the data) and do a much better job of reflecting recent technical developments as well as the existing needs for MS and NMR spectroscopy. These markup language formats were also designed to help address the specific needs of metabolomic researchers in that they can be used to capture more information and annotate pure compound reference spectra, as well as the spectra obtained from complex biofluid mixtures.

Molecular Editors

As seen in Chapter 12, having the right tools to visualize and edit large molecules such as protein, DNA, and RNA structures is critical to gaining insights into their function, binding sites, mechanisms of action, evolution, and overall architecture. The same is true for small molecules. Because metabolite structures are tiny compared with protein or RNA structures, it is often possible to draw metabolites by hand using a type of software program called a “molecular editor.” Molecular editors not only allow users to draw structures, they also allow users to interactively edit, manipulate, and visualize chemical structures. They typically support the reading and writing of one or more standard file formats (such as MOL or SDF) and/or line notations (such as SMILES or InChI). All molecular editors display 2D chemical structures, while some will also support the conversion and display of 3D chemical structures and 3D data formats (such as PDB or PDBx/mmCIF). Most molecular editors are designed with a large central drawing canvas and specialized palettes or structure icons that allow users to

Table 14.1 A list of freely available molecular editors and visualization tools.

Program name	Supplier or reference	Platform(s)	Functions
ACD/ChemSketch	ACD/Labs	Windows, MacOS (VM)	2D drawing, editing, property calculation, logP prediction, structure naming
Avogadro	Hanwell et al. (2012)	Windows, MacOS, Linux, Open Source	3D drawing, editing, 3D visualization
HTML5 Molecular Editor	MolSoft	All, JavaScript	2D drawing, editing
JChemPaint	Krause et al. (2000)	Windows, MacOS, Linux, Open Source	2D drawing, editing, reaction drawing
JME and JSME Molecule Editor	Ertl (2010), Bienfait and Ertl (2013)	All, Java applet, JavaScript	2D drawing, editing
Jmol and JSmol	Hanson et al. (2013)	All, Java applet, JavaScript	3D drawing, visualization
KnowItAll Academic	Bio-Rad	Windows	2D drawing, editing, reaction drawing, spectral analysis, property calculation
MarvinSketch	ChemAxon	All, Java applet	2D drawing, editing, reaction drawing
XDrawChem	www.woodsidelabs.com/chemistry/xdrawchem.php	Windows, MacOS, Linux, Open Source	2D drawing, editing, property prediction, NMR and IR spectra prediction, 3D structure generation

2D, two dimensional; 3D, three dimensional; IR, infrared; NMR, nuclear magnetic resonance.

select, drag, and drop substructures, atoms, or bonds into the canvas at will. Many also allow structure files or SMILES text strings to be dragged into the drawing canvas, instantly rendering them as structures that can be further viewed, manipulated, or saved. Table 14.1 provides a partial list of freely available molecular editors and visualization tools. Some are stand-alone programs while others are available as web-enabled applets. Regardless of what program is chosen, knowing how to use at least one good quality molecular editor is essential for anyone working in a metabolomic laboratory.

Spectral Viewers

In the field of metabolomics, chemical spectra are often just as important as chemical structures. Therefore, having the right tools to display, annotate, and manipulate spectra has been particularly important for the development of metabolomics. Many top-quality spectral viewing tools are sold with modern MS or NMR instruments. There are also numerous independent, third-party commercial suppliers of spectral viewing/manipulation software. As a result, there are relatively few free, stand-alone programs designed for viewing infrared (IR), MS, and NMR spectra. Many of the commercial tools use their own vendor-specific format, but nearly all spectral viewing tools also support a common spectral exchange format called JCAMP-DX (or *.jdx) format. Two freely available spectral viewing tools that use JCAMP-DX are the JCAMP-DX Data Viewer and JDXview (see Internet Resources); these are compatible with the Windows operating systems only. An open source Java version called JSpecView has also been developed (Lancashire 2007) and is now part of JSmol. JSpecView (and JSmol) are easy-to-use, platform-independent tools for spectral viewing, annotation, and manipulation. While JCAMP-DX-compatible viewers still dominate the field, there has also been a push to develop chemical spectral visualization and editing tools that can work with more modern data formats such as mzML and nmrML. Two of the more notable freeware tools include mMass (Niedermeyer 2016), which is a mass spectral processing tool,

and JSpectraViewer (see Internet Resources), which is an NMR spectral analysis tool. The mMass suite is a downloadable, platform-independent package that supports the reading and writing of mzML, mzXML, and mzData formats. It not only allows mass spectral visualization and annotation, but it also supports spectral smoothing, baseline correction, peak picking, and spectral deconvolution. JSpectraViewer is a web-enabled JavaScript tool that allows one-dimensional (1D) NMR spectral visualization and annotation as well as Fourier transformation, phasing, smoothing, baseline correction, and peak assignment. A screenshot from JSpectraViewer for L-alanine is shown in Figure 14.5. JSpectraViewer has been integrated with the Bayesil web server (Ravanbakhsh et al. 2015) to support automated NMR spectral deconvolution analysis of various biofluids. The availability of web-enabled structure and spectral viewing tools or applets such as Jmol, MarvinView, and JSpectraViewer is also having a positive effect on the usability and visualization features offered by a number of metabolomic databases.

Databases

Databases are the cornerstones of bioinformatics. Without databases such as GenBank, UniProt, or the PDB, the fields of genomics or proteomics would not exist. Likewise, the field of metabolomics would not exist without specialized metabolomic databases. Over the past decade, many high-quality metabolomic or chemical compound databases have been developed to address the growing data needs of the metabolomic community. These include the Human Metabolome Database (HMDB; Wishart et al. 2007); PubChem (Wheeler et al. 2006), the Chemical Entities of Biological Interest database (ChEBI; Hastings et al. 2013), LIPID MAPS (Fahy et al. 2007), METLIN (Tautenhahn et al. 2012), the Kyoto Encyclopedia of Genes and Genomes (KEGG; Kanehisa et al. 2014), MetaboLights (Haug et al. 2013), and the Toxic Exposome database (Wishart et al. 2015). These databases can be divided into four broad categories: chemical compound databases, spectral databases, metabolic pathway databases, and organism-specific metabolomic databases. Table 14.2 provides a detailed listing of the main databases in each of these categories and a brief summary of their content. A more detailed explanation of what these databases are and what they contain is given below.

Chemical Compound Databases

Chemical compound databases are searchable databases of chemical names and structures that are intended to provide the broadest possible coverage of the known chemical “space.” As a general rule, chemical compound databases focus more on breadth than on depth. Given their sheer size, essentially all modern chemical compound databases provide support for name/text searching, as well as chemical substructure or fingerprint matching for structure similarity searches. The world’s largest publicly accessible chemical database is PubChem (Wheeler et al. 2006), which is maintained at the U.S. National Center for Biotechnology Information (NCBI). PubChem is an archival database, as it contains data deposited by many different organizations, laboratories, and companies – more than 350 at last count. Currently, PubChem contains more than 80 million unique compounds. Each entry contains chemical structure information, names, synonyms and identifiers, physical properties, and vendor or source information. If available, PubChem entries also include drug and medication information, use and manufacturing data, safety data, toxicity information, literature references, pathway data and biomolecular interactions, and chemical classifications. A set of screenshots from the PubChem database is shown in Figure 14.6. PubChem is extensively linked to PubMed and many compounds in PubChem have descriptions of their biological activity provided through PubMed abstracts. Given its size, accessibility, and high standards, PubChem has become particularly popular among metabolomic researchers. However, it is *very* important to remember that less than 0.1% of the chemicals found in PubChem are actually biological

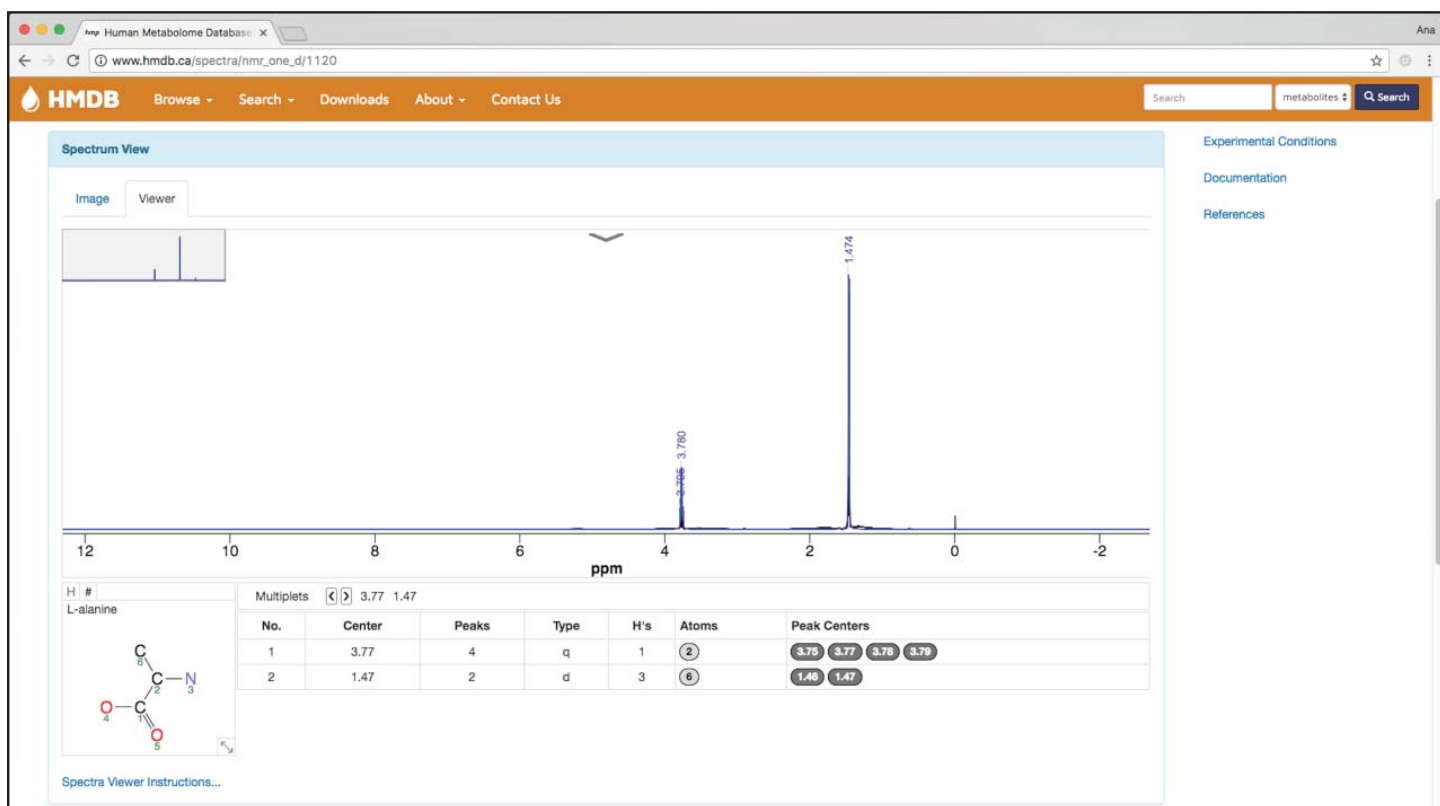


Figure 14.5 The JSpectraViewer image for L-alanine. JSpectraViewer is a Java applet that is also embedded into the Human Metabolome Database. It is displaying the nmrML file shown in Figure 14.4.

Table 14.2 A list of open access chemical, spectral, pathway, and metabolomic databases.

Database name	Database type	Reference	Content
PubChem	Chemical compound DB	Wheeler et al. (2006)	91 million compounds, names, chemical properties, assays, IDs
ChemSpider	Chemical compound DB	Williams (2008)	50+ million compounds, spectra, names, IDs
LIPID MAPS	Chemical compound DB	Fahy et al. (2007)	40 000+ lipid structures, nomenclature, classification
ChEBI	Chemical compound DB	Hastings et al. (2013)	52 000 compounds, nomenclature, ontology
KNAPSAcK	Chemical compound DB	Nakamura et al. (2013)	50 000+ compounds, 111 000 species assignments
NMRShiftDB	Spectral DB	Steinbeck and Kuhn (2004)	43 000+ compounds with NMR spectra
BioMagResBank	Spectral DB	Markley et al. (2008)	900+ compounds with NMR spectra, 4000 NMR spectra
Human Metabolome Database (HMDB)	Spectral DB and metabolomic DB	Wishart et al. (2007)	42 000 human compounds, 105 000 MS spectra, 3800 NMR spectra
MassBank of North America (MoNA)	Spectral DB	Kind et al. (2017)	61 000+ compounds, 211 000 MS spectra
METLIN	Spectral DB	Tautenhahn et al. (2012)	200 000+ compounds, MS/MS data on 10 000+ compounds
Golm Metabolome DB	Spectral DB	Kopka et al. (2005)	26 000+ spectra for 2200+ metabolites
CFM-ID	Spectral DB	Allen et al. (2014)	300 000+ predicted MS spectra for 100 000+ compounds
Kyoto Encyclopedia of Genes and Genomes (KEGG)	Pathway DB	Kanehisa et al. (2014)	18 000+ compounds, 512 metabolic pathways, 4600 organisms
Reactome	Pathway DB	Croft et al. (2011)	2100+ human protein and metabolite pathways
WikiPathways	Pathway DB	Kelder et al. (2012)	2400+ protein and metabolite pathways
Small Molecule Pathway Database (SMPDB)	Pathway DB	Jewison et al. (2014)	724 human metabolite pathways
MetaCyc, BioCyc databases	Pathway DB	Karp et al. (2000)	2500+ metabolite pathways for 2800+ organisms
MetaboLights	Metabolomic DB	Haug et al. (2013)	24 000+ metabolites from 2000+ organisms, 400+ studies
Metabolomics Workbench	Metabolomic DB	Sud et al. (2016)	60 000+ metabolites from 25 organisms, 300+ studies
Yeast Metabolome Database (YMDB)	Metabolomic DB	Jewison et al. (2012)	16 000+ metabolites, 30 000+ MS and NMR spectra
Toxic Exposome Database (T3DB)	Metabolomic DB	Wishart et al. (2015)	3600+ compounds, 11 000+ MS and NMR spectra

DB, database; MS, mass spectrometry; NMR, nuclear magnetic resonance.

compounds. Even fewer are used in industrial manufacturing or have ever been released into the environment. This means that searching through PubChem for compound matches in metabolomic or exposure assessment experiments will lead to a 99.9% false-positive rate.

Of course, PubChem is not the only publicly available chemical compound database. Other, more specialized, chemical databases exist. These often contain different kinds of

NIH | NLM | National Center for Biotechnology Information

PubChem OPEN CHEMISTRY DATABASE

Search Compounds

Compound Summary for CID 5950

L-alanine

STRUC­TURE VENDORS DRUG INFO PHARMACOLOGY LITERATURE PATENTS BIOACTIVITIES

PubChem CID: 5950

Chemical Names: L-alanine; 56-41-7; Alanine; (S)-Alanine; (S)-2-Aminopropanoic acid; L-alpha-alanine More...

Molecular Formula: C₃H₇NO₂

Molecular Weight: 89.094 g/mol

InChI Key: QNAYBMKLOCPYQJ-REOHCLBHSA-N

Drug Information: Drug Indication Therapeutic Uses Clinical Trials FDA UNII

Safety Summary: Laboratory Chemical Safety Summary (LCSS)

L-alanine is a non-essential amino acid that occurs in high levels in its free state in plasma. It is produced from pyruvate by transamination. It is involved in sugar and acid metabolism, increases IMMUNITY, and provides energy for muscle tissue, BRAIN, and the CENTRAL NERVOUS SYSTEM.

Alanine is an Amino Acid. The chemical classification of alanine is Amino Acids.

Alanine is a small non-essential amino acid in humans. Alanine is one of the most widely used for protein construction and is involved in the metabolism of tryptophan and vitamin pyridoxine. Alanine is an important source of energy for muscles and central nervous system, strengthens the immune system, helps in the metabolism of sugars and organic acids, and displays a cholesterol-reducing effect in animals. (NCI04)

PubChem > COMPOUND > L-ALANINE

Create Date: 2004-09-16

(a)

L-Alanine

17 Biomolecular Interactions and Pathways

17.1 Protein Bound 3-D Structures

MIMDB ID: 151457 MIMDB ID: 151656 MIMDB ID: 151196 MIMDB ID: 148754 MIMDB ID: 146412

1 - 5 of 43

17.2 Biosystems and Pathways

BioSystems ID	BioSystems Name
336	Taurine and hypotaurine metabolism
338	Selenocompound metabolism
342	D-Alanine metabolism
395	Carbon fixation in photosynthetic organisms
424	Aminoacyl-tRNA biosynthesis

17.3 DrugBank Interactions

Target: Alanine aminotransferase 2

General Function: Pyridoxal phosphate binding

(b)

Figure 14.6 A selection of two screenshots from the PubChem web pages for the molecule L-alanine. (a) The header data seen in most PubChem entries. (b) The biomolecular interaction data collected for L-alanine. The actual L-alanine entry for PubChem contains many other images, hyperlinks, chemical/biological descriptors, and references.

data not routinely captured by PubChem. ChemSpider (Williams 2008), for example, is a well-regarded, open access chemical database containing more than 30 million compounds. It is particularly known for its carefully curated chemical synonym collection and its extensive collection of spectral data. However, like PubChem, the vast majority of compounds in ChemSpider are not biological or are not found in the environment. Other databases of note include LIPID MAPS (Fahy et al. 2007), a comprehensive database of more than 30 000

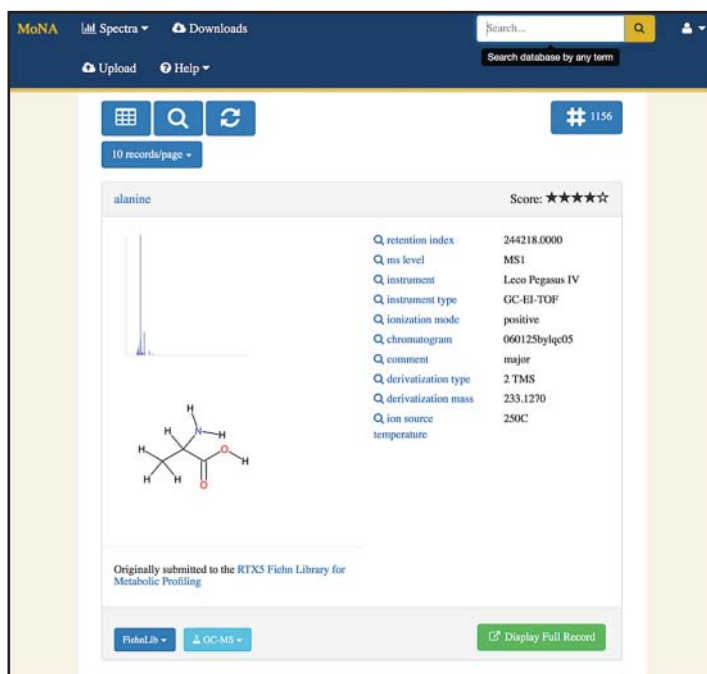
biological lipids; ChEBI (Hastings et al. 2013), a database of 40 000+ biologically interesting compounds; and KNApSACk (Nakamura et al. 2013), a database of nearly 30 000 plant phytochemicals. LIPID MAPS, ChEBI, and KNApSACk are examples of smaller, natural product databases that are generally far more useful to metabolomic researchers than PubChem or ChemSpider.

Spectral Databases

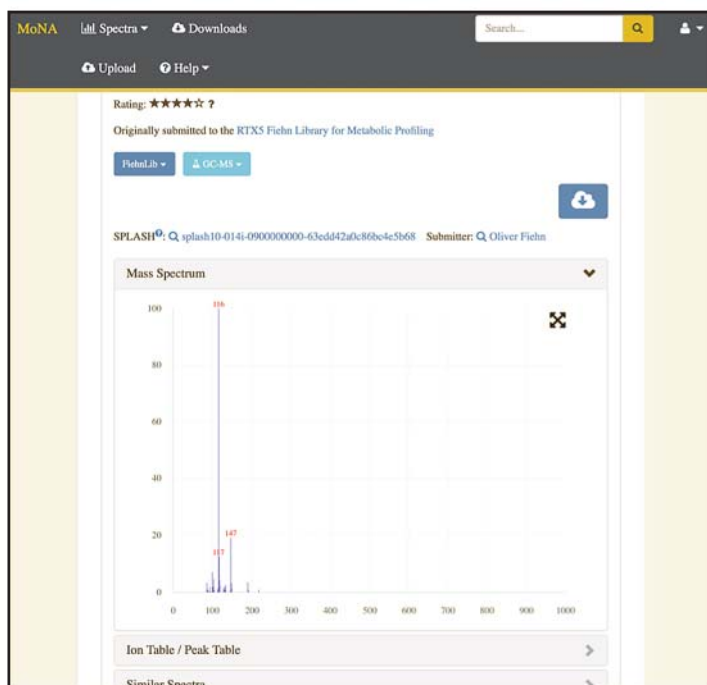
Spectral databases primarily contain experimental 1D NMR, electron ionization (EI)-MS, or electrospray ionization (ESI) tandem mass (MS/MS) spectra of pure chemical compounds. These collections of reference spectra are critical for identifying compounds or confirming a compound's identity. This identification process is particularly important in metabolomics. While there are a number of excellent and very extensive commercial spectral libraries sold by companies such as Wiley, Aldrich, ACD/Labs, and Bio-Rad, there are also a growing number of free, open access spectral databases. Many of these resources support sub-spectral peak searching or global spectral matching as well as standard text querying.

Open access, referential ^1H and ^{13}C NMR spectra at various NMR field strengths can be found in NMRShiftDB and NMRShiftDB2 (Steinbeck and Kuhn 2004), BioMagResBank (Markley et al. 2008), and HMDB (Wishart et al. 2007). NMRShiftDB2 contains nearly 52 000 ^1H and ^{13}C spectra for more than 40 000 compounds. Unfortunately, most of these spectra (>90%) were not collected on biological compounds (i.e. metabolites) and most were not collected in water (which is the standard solvent for most metabolomic experiments). These solvent differences can lead to chemical shift and spectral peak differences, making compound identification via spectral matching somewhat difficult. The BioMagResBank and HMDB contain several thousand high-field (400–700 MHz) NMR spectra for about 1000 common metabolites. Almost all of these spectra are from well-known metabolites and almost all have been collected in water. While the number of reference NMR spectra now available for metabolomics is impressive, this number pales in comparison with the number of EI-MS or ESI-MS/MS spectra that are now publicly available. Literally, hundreds of thousands of ESI-MS/MS and EI-MS spectra can be accessed, viewed, and searched via NIST (the MS database maintained by the U.S. National Institute of Standards and Technology), MassBank of North America (MoNA; Kind et al. 2017), MzCloud, METLIN (Tautenhahn et al. 2012), and the Golm Metabolome Database (Kopka et al. 2005). MoNA is a particularly important resource for metabolomics, as it contains more than 190 000 measured and predicted spectra from more than 80 000 different metabolites. MoNA also supports user deposition of measured MS and MS/MS spectra. A set of screenshots corresponding to the MoNA entry for L-alanine is shown in Figure 14.7.

The challenge with using the spectra from these MS databases is that each compound is often represented by dozens of different MS spectra collected on different MS instruments under different ionization conditions or at different collision energies or with different chemical modifications. So, while the number of *experimentally* collected MS spectra is large, the actual number of unique (parent) compounds represented by this diverse collection is probably less than 30 000. This represents a tiny fraction (perhaps <15%) of known or expected metabolites that has been estimated to be in excess of 200 000 (Psychogios et al. 2011). Given the striking shortage of experimentally collected MS/MS or EI-MS spectra and the diminishing likelihood that existing MS databases will expand by any significant amount in the near term, a number of investigators have started to use computational tools to predict MS/MS and EI-MS spectra with impressive accuracy (Allen et al. 2015, 2016). Many of these *in silico* MS spectra are now available through the Competitive Fragmentation Modeling and Identification (CFM-ID) database (Allen et al. 2014). Regardless of whether the MS spectra are experimentally collected or computationally predicted, MS databases are playing an increasingly important role in compound identification and compound confirmation through their integration with MS spectral processing tools.



(a)



(b)

Figure 14.7 Two screenshots of the gas chromatography–mass spectrometry (GC-MS) data page for L-alanine as it appears in the MassBank of North America (MoNA). (a) The typical results page from a compound query. (b) The expanded GC-MS spectrum for L-alanine.

Metabolic Pathway Databases

Metabolic pathway databases provide a centralized collection of schematic pathways that depict the current state of the knowledge regarding metabolic (meaning catabolic, anabolic, or signaling) processes that occur within a cell, tissue, or organism. In this regard, metabolic

pathway databases play a key role in the biological interpretation and visualization of metabolomic data. Some of the most popular small molecule pathway databases include web-based resources such as KEGG (Kanehisa et al. 2014), the Reactome database (Croft et al. 2011), the “Cyc” databases (Karp et al. 2000), WikiPathways (Kelder et al. 2012), and the Small Molecule Pathway Database (SMPDB; Jewison et al. 2014). A number of commercial pathway databases also exist, such as BioCarta, TransPath (from BioBase, Inc.), and Ingenuity Pathway Analysis (from Ingenuity Systems, Inc.).

Most metabolic pathway databases have been designed to facilitate the exploration of metabolism and metabolites across many different species. This has played a key role in improving our understanding of the evolution and conservation of many aspects of metabolism. Metabolic pathway databases with broad species coverage, such as KEGG and Reactome, tend to use pathway diagrams that are very generic and highly simplified, while those that are more organism specific, such as SMPDB, tend to use pathway diagrams that are much richer in detail, color, and content. Most pathway databases support interactive image mapping with hyperlinked information content that allows users to view chemical information (if a compound is clicked) or brief summaries of genes and/or proteins (if a protein or enzyme is clicked). Almost all pathway databases support some kind of limited text search and a few, such as Reactome, SMPDB, and the Cyc databases, support the mapping of gene, protein, and/or metabolite expression data onto pathway diagrams. Most pathway databases also provide their pathway data in common, machine-readable data exchange formats such as BioPAX (Demir et al. 2010), Systems Biology Markup Language (Hucka et al. 2003), or Systems Biology Graphical Notation Markup Language (van Iersel et al. 2012). Others, such as KEGG, have their own unique dialect or data exchange format (called KGML or KEGG Markup Language). More information about pathway databases can be found in Chapter 13.

Organism-Specific Metabolomic Databases

Modern metabolomic databases typically combine all the features found in compound, spectral, and pathway databases into a single resource. In other words, comprehensive metabolomic databases must be a one-stop shop that supports nearly all aspects of a metabolomic investigation *for a specific organism*. Historically, most metabolomic researchers were so desperate for spectral or compound databases that they did not really care which organism the data were derived from. However, without proper consideration of which organism is being studied, many metabolomic findings and tentative compound identifications are likely incorrect.

There are currently six widely used comprehensive metabolomic databases. Two are archival resources for metabolomic data deposition and four are curated, referential databases designed to cover the metabolomes of specific organisms or specific environments. The two archival databases are the Metabolomics Workbench (Sud et al. 2016), which is maintained at the University of California in San Diego, and MetaboLights (Haug et al. 2013), which is operated by the European Bioinformatics Institute (EBI). MetaboLights and the Metabolomics Workbench are the metabolomic equivalent to the GenBank or PDB databases. They both accept raw and processed metabolomic data and both support metabolomic data analysis. Both resources also mine the deposited data (and other external resources) to provide referential data, such as compound structures, compound names, compound concentrations (if available), and spectral information about individual metabolites. This “reference layer” is of considerable interest to metabolomic researchers, as it provides the necessary data to compare and confirm tentative compound identifications. It also allows researchers to develop predictive tools for metabolomic research and to conduct large-scale metabolic comparisons.

The other set of curated, referential metabolomic databases includes the Human Metabolome Database (HMDB; Wishart et al. 2007), the *E. coli* Metabolome Database (ECMDB; Guo et al. 2013), the Yeast Metabolome Database (YMDB; Jewison et al. 2012), and the Toxic Exposome Database (T3DB; Wishart et al. 2015). The HMDB is a comprehensive

online resource containing referential information about all the known or expected small molecule metabolites found in humans. Four types of data are contained in the database: chemical data, spectral data, clinical data, and molecular biology/biochemistry data. A set of screenshots from the HMDB is shown in Figure 14.8. The latest version of the database contains more than 114 000 compounds, 5700 protein targets, enzymes, or transporters, >18 000 concentration entries, >45 000 pathway diagrams, and >330 000 MS/NMR spectra (both experimental and predicted). The HMDB also has extensive spectral and mass-matching tools to facilitate compound identification, as well as tools for text, sequence, and chemical structure searches. Many of the compounds in the HMDB are endogenous metabolites, but approximately one-quarter of the entries are actually derived from food products (both raw and prepared) that humans consume. Another 3% of the compounds in the HMDB are derived from drugs and drug metabolites.

The ECMDB and YMDB are similar in structure, design, and content to the HMDB. However, both *Escherichia coli* and *Saccharomyces cerevisiae* are somewhat simpler organisms than humans, with smaller genomes and less complex metabolic processes. As a result, the quantity of information in these databases is significantly smaller. In particular, the ECMDB only has data on 3700 compounds, while the YMDB has data on less than 12 000 compounds. However, substantially more is known about microbial metabolism than human metabolism. As a result, the ECMDB has nearly 1600 illustrated metabolic pathways covering nearly 90% of its metabolome, as compared with the HMDB, which has 25 000 pathways covering just 20% of the human metabolome.

In contrast to the other metabolomic databases, the T3DB is an exposome database. The exposome refers to the collection of chemicals (primarily toxic or xenobiotic) to which an organism may be exposed over its lifetime (Wild 2005). In this regard, the T3DB contains comprehensive information on toxic environmental chemicals, such as herbicides, pesticides, pollutants, and certain endogenous toxins such as uremic toxins (which interfere with kidney function) or oncometabolites (associated with cancer). Therefore, the T3DB is not an organism-specific database but, rather, an environment-specific database. Most of the chemicals of concern in the T3DB can be found in (or affect) humans, other mammals, reptiles, amphibians, fish, insects, and plants. The T3DB also contains extensive data on the biological targets, binding constants, mechanisms of toxicity, and toxic concentrations. All of these organism-specific metabolomic databases have spectral- and mass-matching software to facilitate compound identification, as well as tools for text, sequence, and chemical structure searches.

Bioinformatics for Metabolite Identification

The vast majority of metabolomic experiments are conducted as case-control studies, designed to identify causative or predictive biomarkers of disease. In a metabolomic case-control study, NMR and/or MS-based data are collected for a number (10–1000) of normal or healthy control samples and a nearly equal number of “case” (diseased, treated, perturbed) samples. In some situations, there may be two or more case cohorts. Comparing the two (or more) groups and looking for important differences that distinguish between the groups is often the main objective of these kinds of case-control studies. Regardless of how the study is designed, a typical metabolomic experiment will almost always generate an enormous quantity of MS or NMR spectral data (often gigabytes in size). The process of analyzing and interpreting metabolomic data is actually very similar to the process used to analyze or interpret transcriptomic (microarray or RNA-seq) data or proteomic data. All three methods require converting the raw data to lists of “features,” using multivariate statistics to convert the feature lists into shorter lists of significant features, and determining how these significant features are involved in various biological pathways or processes. The later sections of this chapter will describe how these three analysis steps are conducted. This section will focus on metabolite identification as it pertains to both targeted and untargeted metabolomics (Box 14.1).

HMDB The Metabolomics Innovation Centre Quantitative metabolomics services for biomarker discovery and validation.

Showing metabocard for L-Alanine (HMDB0000161)

Identification Taxonomy Ontology Physical properties Spectra Biological properties Concentrations Links References

XML

enzymes (17) transporters (4) Show 21 proteins Show Metabolites with Similar Structures

Record Information


Version	4.0
Creation Date	2005-11-16 15:48:42 UTC
Update Date	2017-09-27 06:59:22 UTC
HMDB ID	HMDB0000161
Secondary Accession Numbers	• HMDB00161

Metabolite Identification

Common Name L-Alanine

Description Alanine is a nonessential amino acid made in the body from the conversion of the carbohydrate pyruvate or the breakdown of DNA and the dipeptides casein and anserine. It is highly concentrated in muscle and is one of the most important amino acids released by muscle, functioning as a major energy source. Plasma alanine is often decreased when the BCAA (Branched Chain Amino Acids) are deficient. This finding may relate to muscle metabolism. Alanine is highly concentrated in meat products and other high-protein foods like wheat germ and cottage cheese. Alanine is an important participant as well as regulator in glucose metabolism. Alanine levels parallel blood sugar levels in both diabetes and hypoglycemia, and alanine reduces both severe hypoglycemia and the ketosis of diabetes. It is an important amino acid for lymphocyte reproduction and immunity. Alanine therapy has helped dissolve kidney stones in experimental animals. Normal alanine metabolism, like that of other amino acids, is highly dependent upon enzymes that contain vitamin B6. Alanine, like GABA, taurine and glycine, is an inhibitory neurotransmitter in the brain. <http://www.dcnutrition.com/AminoAcids/>.

Structure



(a)

Identification Taxonomy Ontology Physical properties Spectra Biological properties Concentrations Links References enzymes (17)

transporters (4) Show 21 proteins XML

Spectra

Spectra Type	Description	Splash Key	
GC-MS	GC-MS Spectrum - GC-EI-TOF (Pegasus III TOF-MS system, Leco; GC 6890, Agilent Technologies) (1 TMS)	splash10-0141-0900000000-c7f6bacc291e83054e	View in MoNA
GC-MS	GC-MS Spectrum - GC-EI-TOF (Pegasus III TOF-MS system, Leco; GC 6890, Agilent Technologies) (Non-derivatized)	splash10-0141-0900000000-381dcd4d9ea77be0b8a5	View in MoNA
GC-MS	GC-MS Spectrum - GC-EI-TOF (Pegasus III TOF-MS system, Leco; GC 6890, Agilent Technologies) (3 TMS)	splash10-01b9-6900000000-6a7c1bb2915e5d0791f1	View in MoNA
GC-MS	GC-MS Spectrum - GC-MS (2 TMS)	splash10-0141-1900000000-84b38982562c29a8148	View in MoNA
GC-MS	GC-MS Spectrum - EI-B (Non-derivatized)	splash10-00k1-9000000000-b2f17507be509a85821b	View in MoNA
GC-MS	GC-MS Spectrum - EI-B (Non-derivatized)	splash10-0141-0900000000-941672891ab64c5015d	View in MoNA
GC-MS	GC-MS Spectrum - EI-B (Non-derivatized)	splash10-0079-0910000000-47b2e3aa274a653a64e	View in MoNA
GC-MS	GC-MS Spectrum - GC-EI-TOF (Non-derivatized)	splash10-0141-0900000000-c7f6bacc291e83054e	View in MoNA
GC-MS	GC-MS Spectrum - GC-EI-TOF (Non-derivatized)	splash10-0141-0900000000-381dcd4d9ea77be0b8a5	View in MoNA
GC-MS	GC-MS Spectrum - GC-EI-QQ (Non-derivatized)	splash10-0a41-1940000000-5de9f7c902aa1e1e507	View in MoNA
GC-MS	GC-MS Spectrum - GC-EI-TOF (Non-derivatized)	splash10-01b9-6900000000-6a7c1bb2915e5d0791f1	View in MoNA
GC-MS	GC-MS Spectrum - GC-MS (Non-derivatized)	splash10-0141-1900000000-84b38982562c29a8148	View in MoNA
Predicted GC-MS	Predicted GC-MS Spectrum - GC-MS (Non-derivatized) - 70eV, Positive	splash10-0006-9000000000-c31f7a2ced8284a740c09	View in MoNA
Predicted GC-MS	Predicted GC-MS Spectrum - GC-MS (1 TMS) - 70eV, Positive	splash10-0006-9100000000-ab365202152d18664d01	View in MoNA
LC-MS/MS	LC-MS/MS Spectrum - Quattro_QQQ 10V, Positive (Annotated)	splash10-0006-9000000000-96b54b2e69a91ab21be08	View in MoNA
LC-MS/MS	LC-MS/MS Spectrum - Quattro_QQQ 25V, Positive (Annotated)	splash10-0006-9000000000-a800830c399aa1097e1a	View in MoNA
LC-MS/MS	LC-MS/MS Spectrum - Quattro_QQQ 40V, Positive (Annotated)	splash10-0006-9000000000-7253c912562200eac231	View in MoNA
LC-MS/MS	LC-MS/MS Spectrum - EI-B (Hitachi RMU-6M) , Positive	splash10-00k1-9000000000-726947da1a5930a49790	View in MoNA

(b)

Figure 14.8 Two screenshots from the Human Metabolome Database (HMDB) entry for L-alanine. (a) The typical metabocard entry for a compound in the HMDB. The actual L-alanine entry contains more than 120 data fields with chemical, biochemical, and biomedical information about this compound. (b) The list of experimental gas chromatography (GC) and/or liquid chromatography–mass spectrometry (LC-MS) spectra associated with L-alanine.

Box 14.1 Targeted Versus Untargeted Metabolomics

There are two distinct approaches to metabolomics. In one approach (called untargeted metabolomics), the compounds are not initially identified. Instead, the (un-named or unidentified) spectral features or spectral peaks are first extracted and statistically analyzed to identify the most significant features or peaks. It is only after the significant features/peaks have been identified that an attempt is made to identify the compounds corresponding to these peaks. In the other approach (called targeted metabolomics), specific compounds are first identified and quantified by carefully analyzing the peaks and their positions or patterns. The resulting list of compounds and concentrations is then analyzed using multivariate statistics to identify the most significant metabolites. In other words, with targeted metabolomics one identifies metabolites in the first step, while in untargeted metabolomics one identifies metabolites in the last step – if at all (Wishart 2011). Typically, untargeted metabolomic approaches are used for metabolite discovery and hypothesis generation, while targeted metabolomic approaches are used for biomarker discovery and hypothesis confirmation.

Both approaches have their advantages and disadvantages. Untargeted metabolomics is highly amenable to automation and generates a non-biased assessment of metabolite data. However, untargeted metabolomics is not very good at providing absolute metabolite quantification, which limits its reproducibility. Furthermore, many “important” features found via untargeted metabolomics cannot be easily identified. In fact, <2% of detected liquid chromatography–mass spectrometry features in untargeted metabolomic studies are typically identified (da Silva et al. 2015). This limits the conclusions that can be drawn and the ability to interpret the data in a biologically meaningful way. In contrast to untargeted metabolomics, targeted metabolomics is focused on compound identification and absolute compound quantification. This makes targeted metabolomics far more reproducible across different laboratories. On the other hand, targeted metabolomics provides a much more limited or more biased view of the metabolome, as only certain pre-selected metabolites are being measured or identified. A typical targeted metabolomic study will generate quantitative data on between 50 and 200 compounds. However, with recent advances in the field, the number of metabolites typically measured in many targeted studies is increasing. As a result, there is a growing preference for using targeted metabolomics over untargeted metabolomics (Wishart 2011).

Levels of Metabolite Identification

Not all metabolites can be identified equally – at least not using metabolomic approaches. According to the Metabolomics Standards Initiative (MSI) (Sumner et al. 2007) there are four levels of metabolite identification: positively identified compounds (level 1), putatively identified compounds (level 2), compounds putatively identified to be part of a compound class (level 3), and unknown compounds (level 4). Positively identified compounds correspond to those chemicals that have a name, a known structure, a Chemical Abstract Services number, or an InChI identifier. To fall into this category, a compound must be identified by two independent and orthogonal parameters (at least for MS) using a purified, authentic standard collected under identical or near identical data collection conditions. These orthogonal parameters include retention time/index + mass spectrum, accurate parent ion mass + MS/MS spectrum, or accurate parent ion mass + isotope abundance pattern. With NMR, an exact match to the ^1H NMR spectrum of the authentic standard (via spectral deconvolution) or a spectral match to an authentic, spiked-in standard is sufficient to reach the level 1 standard. Putatively identified compounds (level 2) correspond to those where only one analytical measurement matches the authentic compound (retention time only or accurate parent ion mass only) or where the compound has a particularly simple NMR spectrum (one or two peaks) that leaves some ambiguity about its true identity. Certainly, if the compound is known to exist in the

biofluid or extract as indicated by numerous literature reports, these putative compound identifications are much stronger and may be considered “near positive.”

The third level of compound identification is typical of many lipids, where the exact structure of the compound cannot be completely determined but it is known to be a specific class of lipid (a phospholipid or triglyceride) or perhaps an ambiguous chemical structure is known (i.e. PC(38:3) – meaning that it is a phosphatidylcholine with two acyl chains having a total of 38 carbons and three unsaturated bonds). The fourth level of compound identification is the “unknown” category. In metabolomics, there are both “known unknowns” and “unknown unknowns.” A “known unknown” corresponds to a metabolite that has been previously described (in the literature or in a database) but that has not yet been positively or putatively identified in the sample of interest. On the other hand, an “unknown unknown” is a truly novel metabolite that has never been described or formally identified in the literature or by anyone else (to the best of one’s knowledge). Often a compound is labeled as an “unknown” simply because the investigator has not been very thorough in their analyses or because their software/database being used for compound identification is inadequate, incomplete, or too small. These unknowns are technically “known unknowns.” When reporting compounds in metabolomic papers, posters, or presentations it is always a good idea to indicate (in a table) the exact level (1, 2, 3, or 4) at which each compound has been identified.

NMR-Based Compound Identification

The standard method for performing metabolite identification in NMR is to use spectral deconvolution. The idea behind spectral deconvolution is to take a complex spectrum and to simplify it into individual spectra of “pure” chemical components. This is illustrated for NMR in Figure 14.9. In metabolomics, spectral deconvolution means taking a spectrum corresponding to a complex chemical mixture (a biofluid such as blood or urine) and reducing it to the spectra of its individual (pure) chemical components. This process typically requires a specially constructed spectral database, as well as specially developed spectral fitting software. The spectral database used in spectral deconvolution should consist of reference spectra of the pure compound(s) that are known or expected to be in the biological sample of interest. These reference spectra must be collected under the exact same conditions (i.e. the same pH, same solvent, same salt, and same temperature) under which the biofluid was analyzed.

As seen in Figure 14.10b (lower image), a typical ^1H NMR spectrum of a biological mixture will consist of hundreds to thousands of sharp peaks. Individual compounds in this mixture

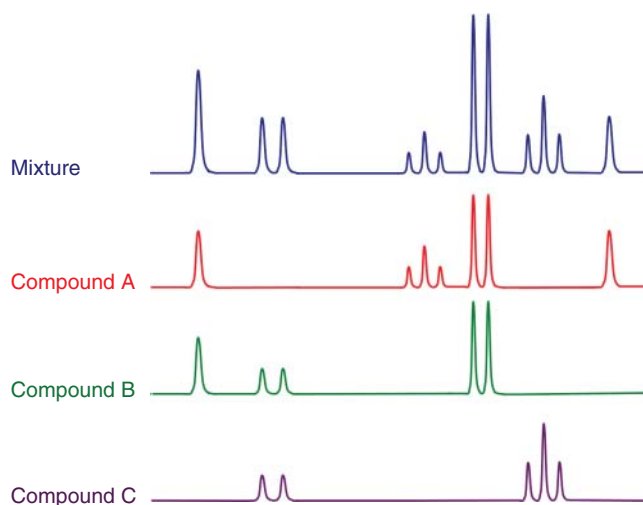
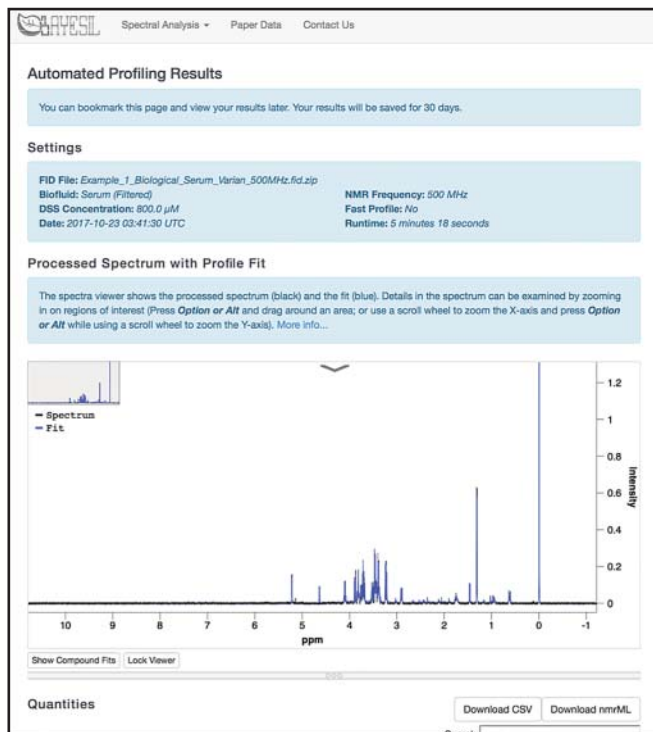
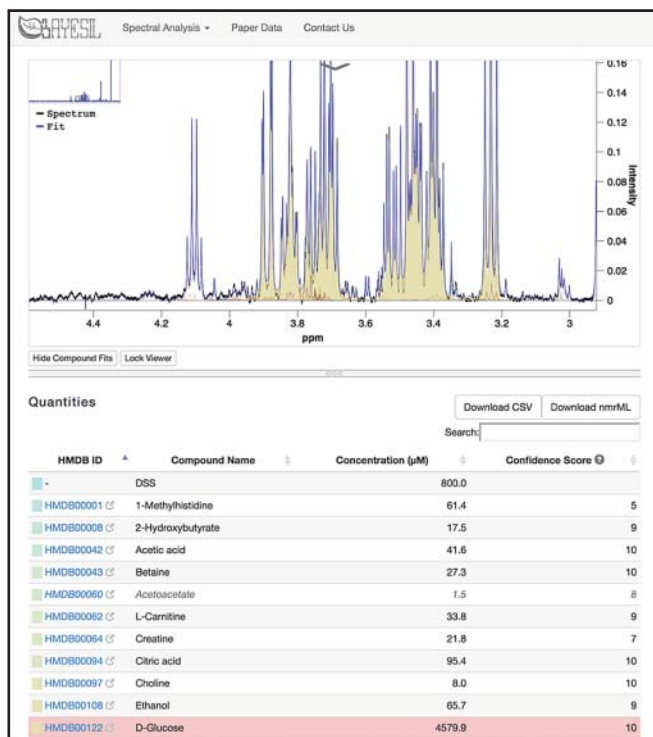


Figure 14.9 A simplified illustration of how spectral deconvolution works for nuclear magnetic resonance (NMR)-based metabolomics. The NMR spectra for compounds A, B, and C are components of the mixture spectrum shown at the top.



(a)



(b)

Figure 14.10 Two screenshots of the Bayesil web server. (a) A nuclear magnetic resonance spectrum of human serum. (b) The spectral fit (and corresponding deconvolution) for glucose, which has been found to have a concentration of 4579.9 μ M in this serum sample.

will consist of an average of 10–15 different peaks or peak clusters (characterized by different intensities, spin couplings, and line shapes) located at different positions throughout the NMR spectrum. By properly matching and fitting a library of individual reference spectra to the observed mixture spectrum, it is possible to simultaneously identify and quantify most compounds in the mixture (Wishart 2008). One reason why spectral deconvolution works particularly well for NMR lies in the fact that most metabolites have unique, almost invariant chemical shift fingerprints made up of multiple compound-specific peaks. The multiplicity of NMR peaks associated with a single compound helps reduce the problem of spectral redundancy. In other words, with NMR it is unlikely that any two randomly selected compounds will have identical numbers of peaks with identical chemical shifts, peak intensities, spin couplings, or line shapes.

There are several commercial programs that support NMR spectral deconvolution for metabolite identification, including AMIX (Bruker) and NMR Suite (Chenomx). Both software packages have large NMR spectral libraries consisting of hundreds of metabolites. Users must manually click, drag, and resize the reference spectra to obtain good spectral fits. Newer versions of these packages now support semi-automatic deconvolution for higher throughput analysis. More recently, Bruker has introduced the WineScreener and JuiceScreener software packages, allowing for fully automated deconvolution of NMR spectra of wines, juices, and even honey. However, this software must be purchased with a specially designed NMR spectrometer, making this a very expensive investment.

In addition to the commercial packages for NMR spectral deconvolution, there are also several freeware packages or web servers that have recently become available. These include Bayesil (Ravanbakhsh et al. 2015) and BATMAN (Hao et al. 2014). BATMAN is a downloadable software package that automatically deconvolutes 1D ^1H NMR spectra using Bayesian statistics. It can both identify and quantify compounds; however, it requires that users must manually phase, reference, and baseline correct their NMR spectra prior to the fitting process. This can lead to significant variation in metabolite quantification from one user to the next. The fitting algorithm used by BATMAN is also quite slow (taking on the order of hours to complete) and is limited to handling mixtures of just 20–25 compounds (which excludes most biofluids). On the other hand, Bayesil is very fast (<2 minutes) and can handle mixtures of up to 60 compounds. Bayesil also automatically performs spectral phasing (adjusting the shape of the NMR peaks so they appear fully above the baseline), chemical shift referencing (defining the 0.00 ppm origin) and baseline correction (making the baseline or peak-free regions perfectly flat), which ensures greater reproducibility and interlaboratory consistency. The Bayesil web server supports automated deconvolution of serum, plasma, saliva, cerebrospinal fluid, and fecal water at multiple NMR spectrometer frequencies (500, 600, 700 MHz). A set of screenshots for the Bayesil web front end is shown in Figure 14.10.

GC-MS-Based Compound Identification

The process of spectral deconvolution for GC-MS and LC-MS is illustrated in Figure 14.11. As can be seen from this image, a typical GC-MS spectrum or total ion chromatogram (TIC) from a metabolite mixture will consist of dozens of sharp peaks (corresponding to ion counts) covering an elution time of about 30–45 minutes. Each peak may consist of one or more EI (electron ionization) mass spectra arising from one or more compounds (Figure 14.11). A variety of commercial GC-MS deconvolution tools such as the Automated Mass Spectral Deconvolution and Identification System (AMDIS), Deconvolution and Reporting Software - DRS (Agilent), ChromaTOF (Leco), and AnalyzerPro (SpectralWorks) can be used to deconvolute GC-MS and EI-MS spectra. Once the EI-MS spectra are extracted, metabolite identification is performed in a similar manner to what is done for NMR. Namely, the extracted EI-MS spectra from the mixture are compared, one at a time, with spectral reference libraries containing the EI-MS spectra of thousands of pure, derivatized, and authenticated compounds. EI-MS spectra usually consist of multiple m/z peaks of varying intensity or abundance. Unlike NMR spectra, which have

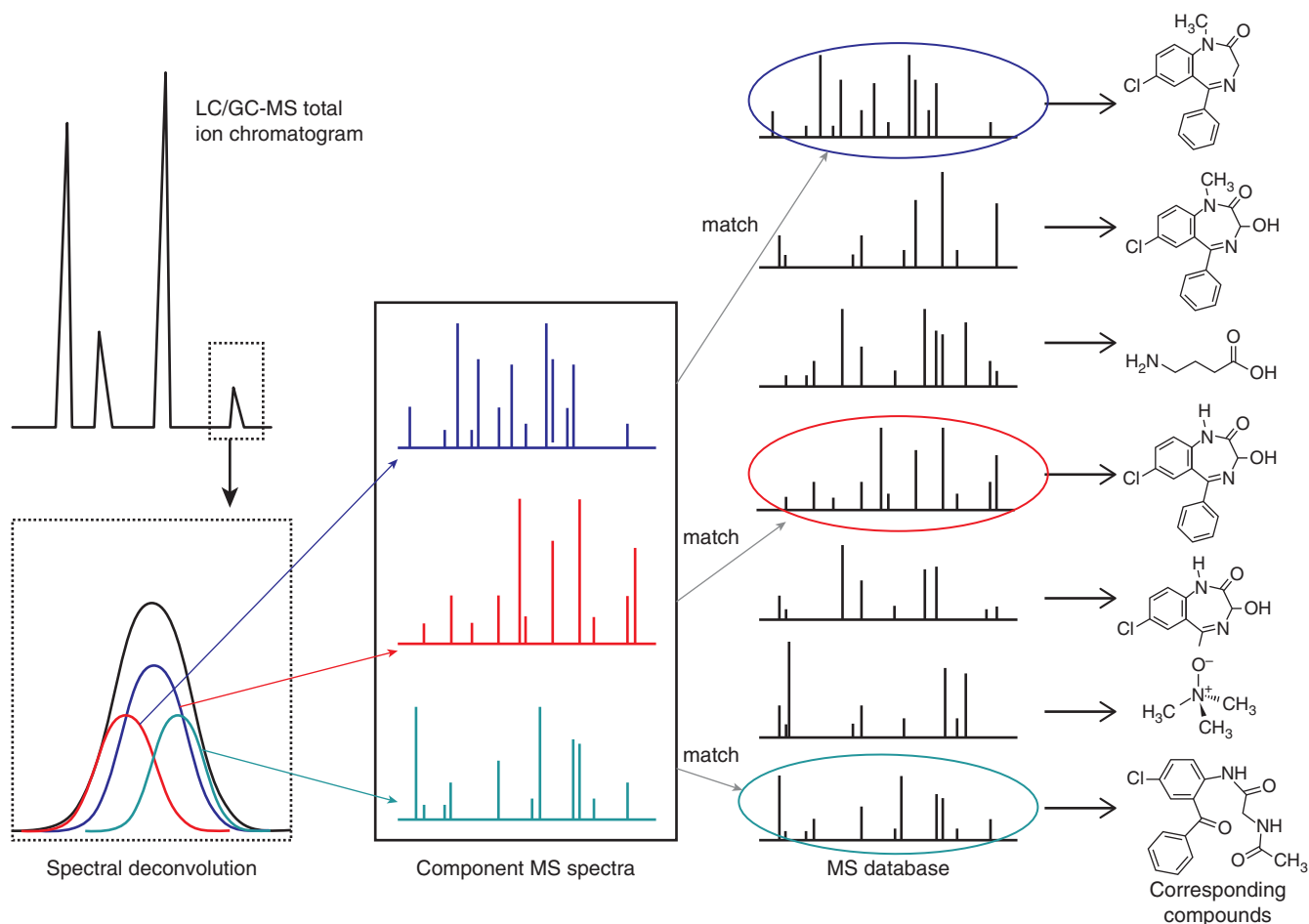


Figure 14.11 An illustration of how spectral deconvolution works for gas chromatography (GC) and/or liquid chromatography–mass spectrometry (LC-MS)-based metabolomics. Peaks are extracted from the chromatogram and the MS, electrospray ionization (ESI)–tandem mass (MS/MS), or electron ionization (EI)-MS spectra are then compared against a library of known compound spectra (in an MS or MS/MS database) to identify the compounds.

characteristic peak shapes and multiplet (multi-peak) patterns, MS spectra can be regarded as single lines or thin bars corresponding to a mass and an intensity. Therefore, the similarity of a query MS spectrum to a reference MS spectrum can be assessed more simply using a term called a match factor (MF), which is defined as the normalized, mass-weighted product of the intensities of the query spectrum and the reference spectrum, per the following equation:

$$MF = \frac{1000 * (\sum wM[I_{qry}I_{ref}]^{1/2})^2}{\sum I_{qry}M * \sum I_{ref}M}$$

Here, I_{ref} corresponds to the intensities of the reference spectra, I_{qry} corresponds to the intensities of the query spectra, M corresponds to the masses (m/z), and w is a weighting term used to penalize uncertain peaks (Stein 1999). As a general rule, a tentative match between EI-MS spectra requires a score of >600 on a scale of 0–1000, with 1000 being a perfect match.

There are three key factors for compound identification by GC-MS: the quality of the extracted query spectrum, the quality or sophistication of the spectral matching algorithm, and the quality and comprehensiveness of the reference spectral database. The quality of the query spectrum is a function of both the instrument (its column, sensitivity, and separation parameters) and the spectral deconvolution software. Assuming the instrumental conditions are optimized, a key issue is often how well the deconvolution software performs. Unlike NMR, where false-positive peaks are extremely rare, GC-MS is frequently plagued with an abundance of false-positive peaks. In some cases, up to 50% of features seen in GC-MS spectra

are fragments, adducts, or derivatives of the column matrix, the derivatization reagents, or of the metabolites themselves. An interesting study (Lu et al. 2008) compared three of the most common GC-MS deconvolution packages (AMDIS, ChromaTOF, and AnalyzerPro) using a defined mixture of 35 compounds with widely varying concentrations. These authors found that both the AMDIS and ChromaTOF packages produced unusually high numbers of false positives or false/impure spectra, while the AnalyzerPro package generally performed best.

Ultimately, the main factors driving the success (or lack of success) in compound identification by GC-MS are the size and quality of the spectral reference database. The most common and widely used resource is NIST's mass spectral database. The latest release contains EI-MS spectra for nearly 200 000 compounds or derivatized compounds, along with retention index values for another 21 800 compounds. However, most of the NIST compounds are not metabolites or are not from biological materials. This can lead to a number of false-positive identifications, especially if authentic standards are not used to verify the identity of the compound. Other databases, albeit somewhat smaller in size, are potentially more suitable for metabolite identification. These include the Golm Database (Kopka et al. 2005), the Fiehn Metabolome Database (BinBase), and the HMDB (Wishart et al. 2007). All of these databases provide retention index data and EI-MS data in a format that is AMDIS compatible. The Golm database is primarily oriented toward plants, while BinBase and the HMDB are oriented more toward mammals.

LC-MS-Based Compound Identification

As seen in Figure 14.11, a typical LC-MS spectrum from a metabolite mixture will consist of many sharp peaks (corresponding to ion counts) covering an elution time of about 10–35 minutes. Each peak may consist of one or more ESI m/z values arising from one or more compounds. As a result, LC-MS metabolomic studies can easily generate a huge number of spectral features or putative compounds (>10 000). This is many times more than what is seen by NMR or GC-MS. Many of these LC-MS features turn out to be noise peaks, column contaminants, in-source fragments, adducts, and isotopic variants. As a result, LC-MS data typically require a considerable amount of post-processing and peak consolidation to reduce the number of peaks to a reliable, countable number (preferably <2000 putative compounds).

LC-MS data are often further complicated by the fact that liquid chromatographic data are substantially more variable from run to run than NMR or GC-MS data. As a result, metabolomic data acquired via LC-MS techniques typically require additional de-noising, alignment, and averaging to ensure that the correct peaks are being picked and compared. Further data reduction is often done using more sophisticated statistical methods (described in Multivariate Statistics) to select for only the most significantly altered peak features. This kind of spectral processing requires sophisticated software that either comes bundled with the LC-MS instrument or that is designed, written, and distributed by highly specialized MS laboratories. Examples of some of the instrument-specific tools include Mass Frontier (ThermoFisher), MassHunter (Agilent), XCMS-Plus (Sciex), ProfileAnalysis (Bruker), Progenesis (Nonlinear Dynamics), and MassLynx (Waters). There are also a number of platform-independent freeware systems, including XCMS (Smith et al. 2006), MS-DIAL (Tsugawa et al. 2015), and MZmine (Katajamaa et al. 2006). All of these software packages support chromatographic and MS spectral alignment, peak finding, calculation of multivariate statistics (for data reduction), parent ion mass matching, molecular formula calculation, and MS/MS spectral matching.

As an LC-MS experiment is being conducted, it is possible to take each parent ion (such as those shown in Figure 14.11) and conduct a further MS fragmentation step to produce an MS/MS (tandem mass) spectrum for that parent ion. This is usually done using a tandem mass spectrometer such as a triple quadrupole (QqQ), quadrupole time-of-flight (QTOF), or an Orbitrap instrument. So, depending on how an LC-MS instrument is configured and how the LC-MS (or LC-MS/MS) data are collected, one can either attempt to identify metabolites

using accurate mass measurements of the parent ions (alone) or one can attempt to identify metabolites by matching MS/MS fragment patterns to appropriate MS/MS spectral libraries.

Metabolite identification via accurate parent ion mass (or, more correctly, the mass-to-charge ratio m/z) measurement requires the use of very-high-resolution MS instruments such as QTOFs, Orbitraps, or Fourier transform ion cyclotron resonance (FT-ICR) spectrometers. If a parent ion mass is measured to four or five decimal places, corresponding to a mass accuracy of <5 ppm, it is usually possible to determine the ion's molecular formula and its putative identity (level 3 identification) through a chemical formula calculator. Several commercial MS chemical formula calculators exist; these include SigmaFit (Bruker), Formula Predictor (Shimadzu), and MassHunter (Agilent), as well as a number of freeware packages such as 7-Golden-Rules (Kind and Fiehn 2007) and SIRIUS (Böcker et al. 2009). By including restrictions on the types of elements typically found in metabolites (i.e. C, N, O, S, H, and P), as well as requirements on hydrogen/carbon ratios, isotopic abundances, and several other expert-driven rules, it is often possible to reduce the number of feasible chemical formulae even further, often by a factor of 15 or more (Kind and Fiehn 2010). Unfortunately, even with these improvements, parent ion-based metabolite identification is still very risky, as there are often many masses or molecular formulae that can still match dozens of metabolites in existing compound databases.

The preferred route of metabolite identification for most LC-MS metabolomic researchers is to use both parent ion (or formula) matching and MS/MS spectral matching. The MS/MS spectrum, with its characteristic fragmentation patterns, provides very useful information about the molecule and its chemical structure. Successful LC-MS/MS spectral matching is critically dependent on having instrument-specific or condition-specific MS/MS product ion fragment libraries. Many of these libraries are bundled with the instrument-specific software packages mentioned earlier. On the other hand, public MS/MS databases, such as METLIN (Tautenhahn et al. 2012), MoNA, and the HMDB (Wishart et al. 2007) are normally used by the freeware packages (XCMS, MS-DIAL, and MZmine) to perform their MS/MS spectral matching. Even with the best spectral databases and the best spectral processing tools, it is still quite difficult to confidently identify (MSI level 2) and partially quantify more than 200–300 metabolites via untargeted LC-MS-based metabolomics. Targeted LC-MS-based metabolomics (which uses multiple reaction monitoring and isotopic dilution analysis) typically allows one to identify and accurately quantify about 150–250 metabolites.

Multivariate Statistics

Targeted metabolomics can easily generate dozens to hundreds of metabolites for each sample run, while untargeted metabolomics can easily generate thousands of features or peaks for each sample run. Regardless of the approach used, metabolomic experiments generate enormous lists consisting of thousands of variables – not unlike proteomic or transcriptomic experiments. As a result, metabolomic researchers often turn to the types of computer tools and computer-based statistics commonly used in proteomics or transcriptomics. As each biofluid sample typically has hundreds to thousands of variables (i.e. metabolites, metabolite concentrations, or peak values) associated with it, the statistical techniques that must be used are called multivariate statistics. In multiple variable or multivariate statistics, the variables are called “dimensions.” One of the primary objectives of multivariate statistics is to reduce the number of variables or dimensions so that the problem can be tackled more simply using traditional univariate statistics such as Student's t -tests or analysis of variance (ANOVA) techniques (see Chapter 18). In particular, multivariate statistics uses a class of mathematical techniques called dimensionality reduction to make multivariate data look more like univariate data. Dimensionality reduction allows one to identify the key components in a large multivariate dataset that contain the maximum amount of information or maximize the differences among groups. As a result, dimensionality reduction reduces a long list of metabolites (or genes or

proteins) to a shorter list of the most significant metabolites (or genes or proteins). The most common form of dimensionality reduction is called principal component analysis (PCA).

Principal Component Analysis

PCA is an unsupervised clustering technique. Clustering is the process of grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups. Clustering helps distinguish groups, such as cases and controls, from one another based on their metabolic, genomic, or proteomic parameters. PCA, which is also known as singular value decomposition or eigenvector analysis, performs clustering based on correlated features in the data. PCA can be easily conducted using a variety of free or nearly free software programs such as MATLAB or the R programming language using R's *prcomp* or *princomp* commands. PCA can also be performed using freely available, downloadable software packages such as XCMS (Smith et al. 2006), MS-DIAL (Tsugawa et al. 2015), MAVEN (Melamud et al. 2010), and Galaxy-M (Davidson et al. 2016), all of which are frequently used for processing LC-MS data. PCA can be performed using MVAPACK (Worley and Powers 2014) to process NMR data. Freely available web servers are also available that support PCA and other common multivariate statistical techniques. These include the Meta-P server (Kastenmüller et al. 2011), MeltDB (Kessler et al. 2013), and MetaboAnalyst (Xia et al. 2015). These web servers provide easy-to-use graphical interfaces that allow users to simply point and click to perform complex multivariate statistical operations or to generate colorful, interactive graphs or tables. MetaboAnalyst is particularly popular in the metabolomic community, with nearly one-third of all published metabolomic papers using this freely available web server.

PCA is a statistical method that determines an optimal linear transformation for a collection of data points such that the properties of that sample are most clearly displayed along a small number of coordinate (or principal) axes. PCA allows metabolomic researchers to easily plot, visualize, and cluster multiple lists of metabolites and their concentrations based on linear combinations of their shared features. A simplified visual explanation of PCA is given in Figure 14.12. Here we use the analogy of projecting shadows on a wall using a flashlight to find a “maximally informative projection” for a particular object. In this example, we are trying to reduce a 3D object into a series of maximally informative 2D projections that would allow us to reconstruct a proper model of the original object. If the object of interest is a thick ring or torus, then by shining the flashlight directly on the face of the ring, a characteristic “ring” shadow would be generated. On the other hand, if the flashlight were directed at the edge of the ring, the resulting shadow would be a less informative sausage-like shape. This sausage shadow, if used alone, would likely lead the observer to the wrong conclusion about what the object was. However, by combining the ring shadow with the sausage shadow (i.e. the two principal components) it is possible to reconstruct the shape and thickness of the original 3D

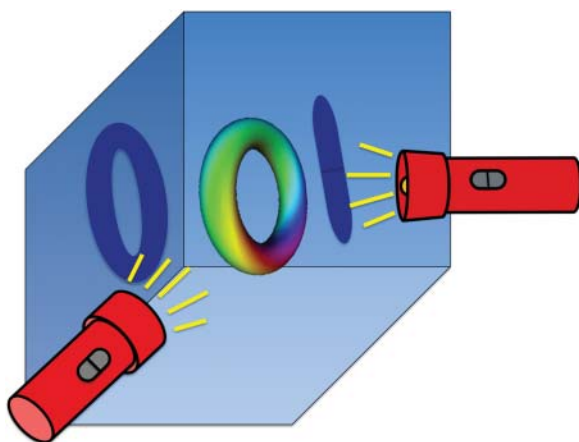


Figure 14.12 An illustration of how principal component analysis can be thought of using a simplified flashlight projection analogy.

ring or torus. While this example shows how a 3D object can have its key components reduced to two dimensions, the strength of PCA is that it can also do the same with a hyperdimensional (multidimensional) object.

PCA is most commonly used in metabolomics to determine whether one or more samples are different from another. It also allows one to identify which variables contribute most to this difference, and whether those variables contribute in the same way (i.e. are correlated) or independently (i.e. uncorrelated) from each other. PCA is particularly appealing because it allows one to visually detect sample clusters or groupings. In particular, the results of a PCA are usually discussed in terms of *scores* and *loadings*. The scores represent the original data in the new coordinate system and the loadings are the weights applied to the original data during the projection process. Plotting out the data using two sets of scores (one for the X -axis and one for the Y -axis) will produce a “scores” plot.

An example of a 3D PCA scores plot generated using MetaboAnalyst is shown in Figure 14.13 (Xia et al. 2015). To understand how this image was generated, we will briefly outline the process. To start, go to the MetaboAnalyst home page (see Internet Resources). At the top of the home page, select *Click here to start*, then click on the *Statistical Analysis* button in the upper left corner of the Module Overview page (Figure 14.14). Then, scroll down the Data Upload page to locate the *Try our test data* section on the bottom half of the page. Use the radio button to select the second concentration dataset (labeled *Metabolite concentrations of 39 rumen...*), then click the *Submit* button at the bottom of the page (Figure 14.15). This action loads the dataset into MetaboAnalyst. After skipping the data integrity check, navigate to the Data Normalization page. This page allows users to scale and normalize the data so that they are more amenable to standard statistical analyses. For this particular dataset, select the Normalization by a pooled sample from the group option for normalization and select group 0 chosen from the pull-down menu. Leave the Data transformation set to *None* and the data scaling as *Auto scaling* (Figure 14.16). Click the *Normalize* button at the bottom of the page, then click on the *View Result* button. The result of these normalization and scaling procedures is shown as a pop-up window (Figure 14.17). Notice how the concentration data that were previously very “skewed” (on the left) are now looking more bell shaped (i.e. Gaussian) in the distribution on the right. Transforming the data to look like this is important so that standard statistical analyses can be performed. After viewing the result, clear the pop-up window and click the *Proceed* button (Figure 14.16). Once these early stage data processing steps are complete, it is possible to start doing the PCA analysis.

MetaboAnalyst contains 16 statistical methods that support metabolomic data analysis. These statistical methods are organized into five categories: univariate analysis, multivariate

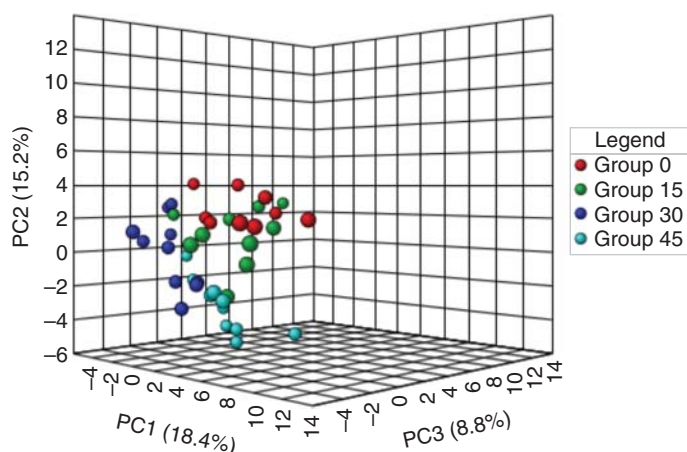


Figure 14.13 A three-dimensional principal component analysis (PCA) “scores” plot showing the separation achieved from analyzing the ruminal fluid from four different groups of cows fed four different diets. The percentage contribution (to an explanation of the variance) of each principal component is labeled on each of the three axes.

MetaboAnalyst 3.0
– a comprehensive tool suite for metabolomic data analysis

Please choose a functional module to proceed:

- Statistical Analysis**
This module offers various commonly used statistical and machine learning methods including t-tests, ANOVA, PCA, PLS-DA and Orthogonal PLS-DA. It also provides clustering and visualization tools to create dendrograms and heatmaps as well as to classify based on random forests and SVM.
- Enrichment Analysis**
This module performs metabolite set enrichment analysis (MSEA) for human and mammalian species based on several libraries containing ~6300 groups of metabolite sets. Users can upload either 1) a list of compounds, 2) a list of compounds with concentrations, or 3) a concentration table.
- Pathway Analysis**
This module supports pathway analysis (integrating enrichment analysis and pathway topology analysis) and visualization for 21 model organisms, including Human, Mouse, Rat, Cow, Chicken, Zebrafish, Arabidopsis thaliana, Rice, Drosophila, Malaria, S. cerevisiae, E.coli. and others, with a total of ~1600 metabolic pathways.
- Time-series/Two-factor Design**
This module supports temporal and two-factor data analysis including data overview, two-way ANOVA, and empirical Bayes time-series analysis for detecting distinctive temporal profiles. It also supports ANOVA-simultaneous component analysis (ASCA) to identify major patterns associated with each experimental factor.
- Power Analysis**
This module uses pilot data to calculate the minimum number of samples required to detect a statistically significant difference between two populations with a given degree of confidence (called Power Analysis).
- Biomarker Analysis**
This module performs various ROC curve based biomarker analyses for a single or multiple biomarkers. It also allows users to manually specify biomarker models as well as new sample prediction.
- Integrated Pathway Analysis**
This module performs integrated metabolic pathway analysis on results obtained from combined metabolomics and gene expression studies conducted under the same experimental conditions.
- Other Utilities**
This module contains several common utility functions. At this moment, **compound ID conversion**, **batch effect correction** and **lipidomics data analysis** are available.

Navigation Sidebar:
[Home](#)
[Overview](#)
[Data Formats](#)
[FAQs](#)
[Tutorials](#)
[Troubleshooting](#)
[Resources](#)
[Update History](#)
[User Stats](#)
[About](#)

Logos:
 GenomeCanada
 GenomeQuébec
 NSERC CRSNG
 TMIC

Figure 14.14 The MetaboAnalyst Module Overview page. This page allows users to select the analysis modules to process or visualize their data.

analysis, significant feature identification, cluster analysis, and classification and feature selection. Multivariate methods include PCA and partial least squares discriminant analysis (PLS-DA), among others. To perform PCA on the dataset under consideration, click the *PCA* hyperlink on the screen that appeared after clicking the *Proceed* button (located under the Chemometrics Analysis banner). After a few seconds, the PCA results should be presented in a multi-panel page. The default panel shows a pairwise scores plot between the first five PCs. The variance explained by each PC is shown on the corresponding diagonal cell. Click the *2D Scores Plot* tab located at the top of the page to get a more detailed score plot. The

MetaboAnalyst 3.0
- a comprehensive tool suite for metabolomic data analysis

1) Upload your data

Tab-delimited text (.txt) or comma-separated values (.csv) file:

Data Type: Concentrations Spectral bins Peak intensity table

Format: Samples in rows (unpaired)

Data File: Choose File no file selected

Submit

Zipped Files (.zip) :

Data Type: NMR peak list MS peak list MS spectra

Data File: Choose File no file selected

Pair File: Choose File no file selected

Submit

2) Try our test data :

Data Type	Description
<input type="radio"/> Concentrations	Metabolite concentrations of 77 urine samples from cancer patients measured by 1H NMR (Elsner R, et al.). Group 1- cachexic; group 2 - control
<input checked="" type="radio"/> Concentrations	Metabolite concentrations of 39 rumen samples measured by proton NMR from dairy cows fed with different proportions of barley grain (Ametaj BN, et al.). Group label - 0, 15, 30, or 45 - indicating the percentage of grain in diet.
<input type="radio"/> NMR spectral bins	Binned 1H NMR spectra of 50 urine samples using 0.04 ppm constant width (Psihogios NG, et al.) Group 1- control; group 2 - severe kidney disease.
<input type="radio"/> NMR peak lists	Peak lists and intensity files for 50 urine samples measured by 1H NMR (Psihogios NG, et al.). Group 1- control; group 2 - severe kidney disease.
<input type="radio"/> Concentrations (paired)	Compound concentrations of 14 urine samples collected from 7 cows at two time points using 1H NMR (unpublished data). Group 1- day 1, group 2- day 4.
<input type="radio"/> MS peak intensities	LC-MS peak intensity table for 12 mice spinal cord samples (Saghatelian et al.). Group 1- wild-type; group 2 - knock-out.
<input type="radio"/> MS peak lists	Three-column LC-MS peak list files for 12 mice spinal cord samples (Saghatelian et al.). Group 1- wild-type; group 2 - knock-out.

Figure 14.15 The MetaboAnalyst Data Upload page. This page allows users to upload their own data or select which test dataset they will use. In this particular example, the second set of data listed in the *Try our test data* has been selected.

default is PC1-PC2 (Figure 14.18). These two components account for >70% of the variation in the samples. We can see the main direction of separation among groups 0, 15, 30, and 45. Groups 0 and 45 are well separated, while group 30 overlaps significantly with both group 15 and group 45. Clicking the *3D Scores Plot* tab will generate the image shown in Figure 14.13. In certain cases, PCA will not succeed in identifying any obvious groupings no matter how many PCs are used. If this is the case, it is wise to accept the result and assume that the presumptive classes or groups cannot be distinguished. Generally speaking, if a PCA analysis

MetaboAnalyst 3.0
— a comprehensive tool suite for metabolomic data analysis

Normalization overview:

The normalization procedures are grouped into three categories. The sample normalization allows general-purpose adjustment for differences among your sample; data transformation and scaling are two different approaches to make individual features more comparable. You can use one or combine them to achieve better results.

Sample normalization

- None
- Sample-specific normalization (i.e. weight, volume) [Click here to specify](#)
- Normalization by sum
- Normalization by median
- Normalization by reference sample (PQN)
- Normalization by a pooled sample from group
- Normalization by reference feature
- Quantile normalization

Data transformation

- None
- Log transformation (generalized logarithm transformation or glog)
- Cube root transformation (take cube root of data values)

Data scaling

- None
- Mean centering (mean-centered only)
- Auto scaling (mean-centered and divided by the standard deviation of each variable)
- Pareto scaling (mean-centered and divided by the square root of standard deviation of each variable)
- Range scaling (mean-centered and divided by the range of each variable)

Normalize View Result Proceed

Figure 14.16 The MetaboAnalyst Data Normalization page. The optimal normalization and scaling operations have been selected.

fails to achieve even a modest group separation, then it is probably not worthwhile using other statistical techniques to try to separate them.

PCA is also a very helpful technique for quantifying the amount of useful information contained in the data. This is typically done by plotting the weightings of the individual components in a PCA “loadings” plot. To generate a loadings plot via MetaboAnalyst, one can use exactly the same process outlined above but instead of clicking on the *2D Scores Plot* (which generated Figure 14.18), one should click on the *Loadings Plot* tab, which shows the loadings for PC1 and PC2 (Figure 14.19). Note that the direction of separation in the original scores plot

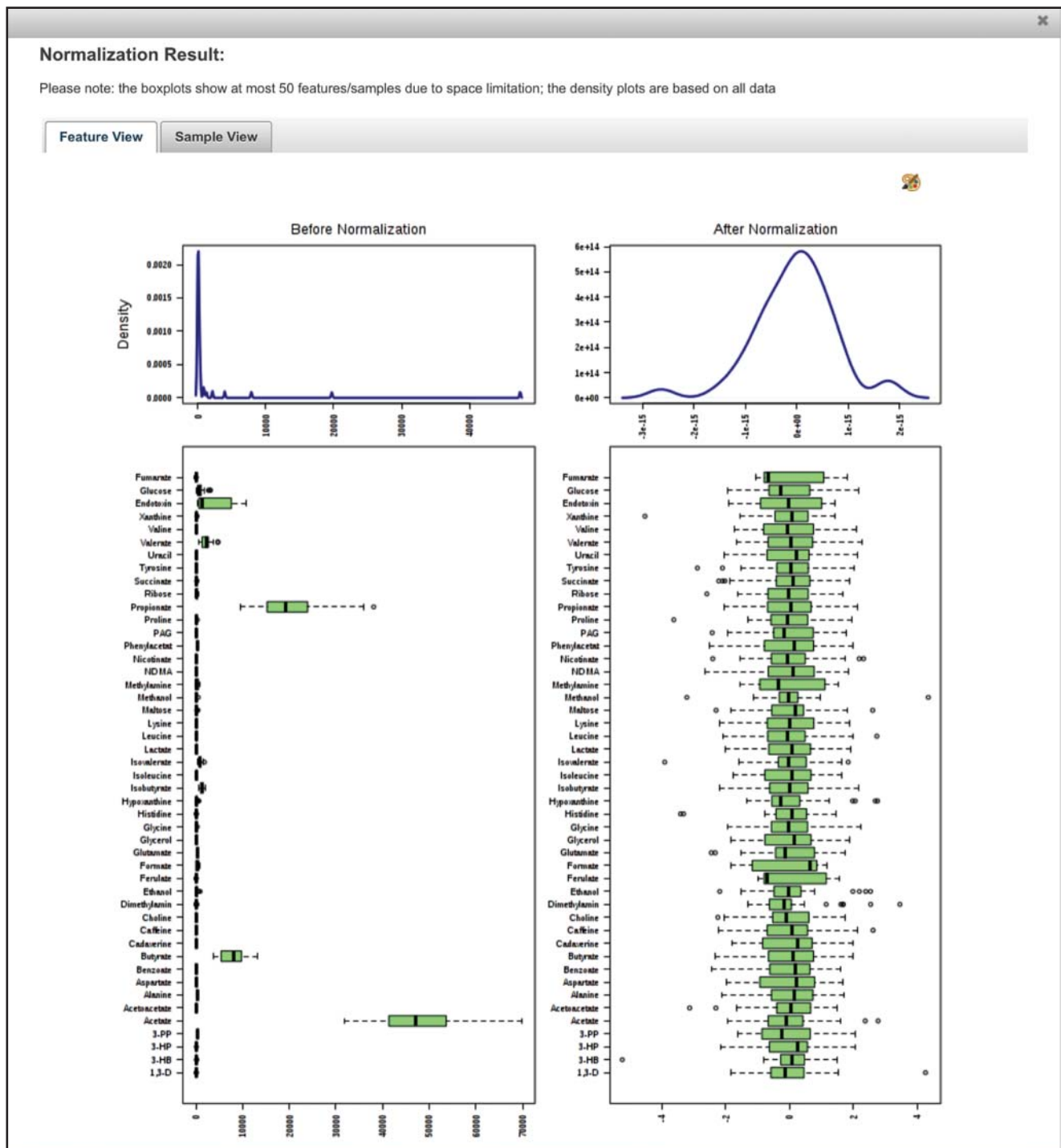


Figure 14.17 The MetaboAnalyst Data Normalization and Scaling results, generated after pressing the *View Result* button at the bottom of the Data Normalization page shown in Figure 14.16. The pop-up window can be cleared (by clicking on the *X* in the top corner) and alternative normalization or scaling functions can be applied. Try to see if you can find a combination of scaling/normalization functions that performs better than the one suggested in the text.

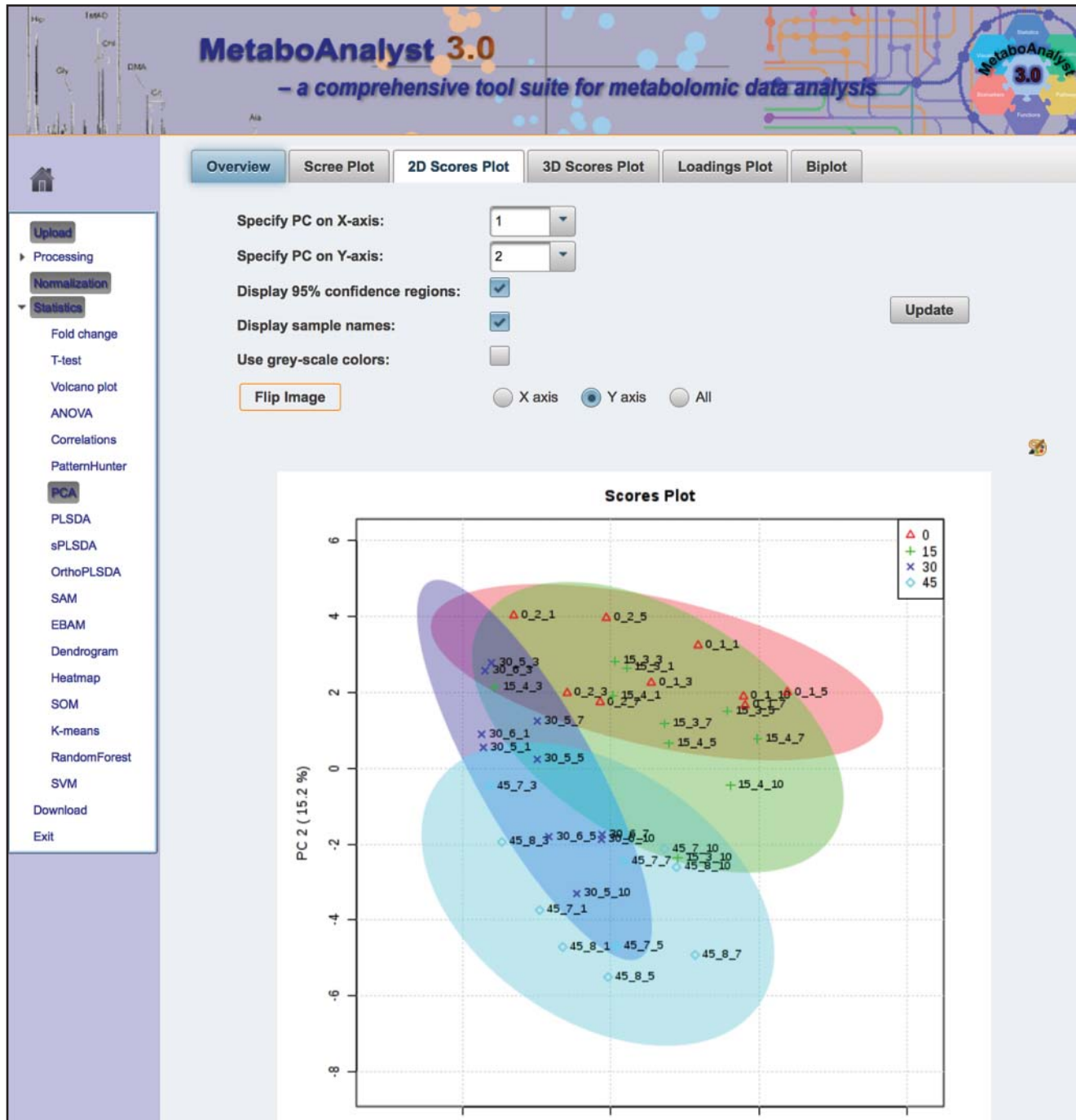


Figure 14.18 A two-dimensional principal component analysis (PCA) “scores” plot showing the separation achieved from analyzing the ruminal fluid from four different groups of cows fed four different diets. The percent contribution (to an explanation of the variance) of each principal component (PC) is labeled on each of the two axes. See text for details.

was from the lower left to the upper right (diagonally). By looking at the compounds in the loadings plot located in the upper right and lower left, the most influential compounds that drive the separation can be identified. This can be done by clicking on individual points in this graph, which produces a box plot in the upper right of the plot. In this example, aspartate, isobutyrate, and 3-phenylpyruvate located on the top right, and endotoxin, glucose, and methylamine located on the bottom left are the key metabolites driving this separation. Note that this kind of loadings plot, where specific metabolite identities are displayed, is only possible if the

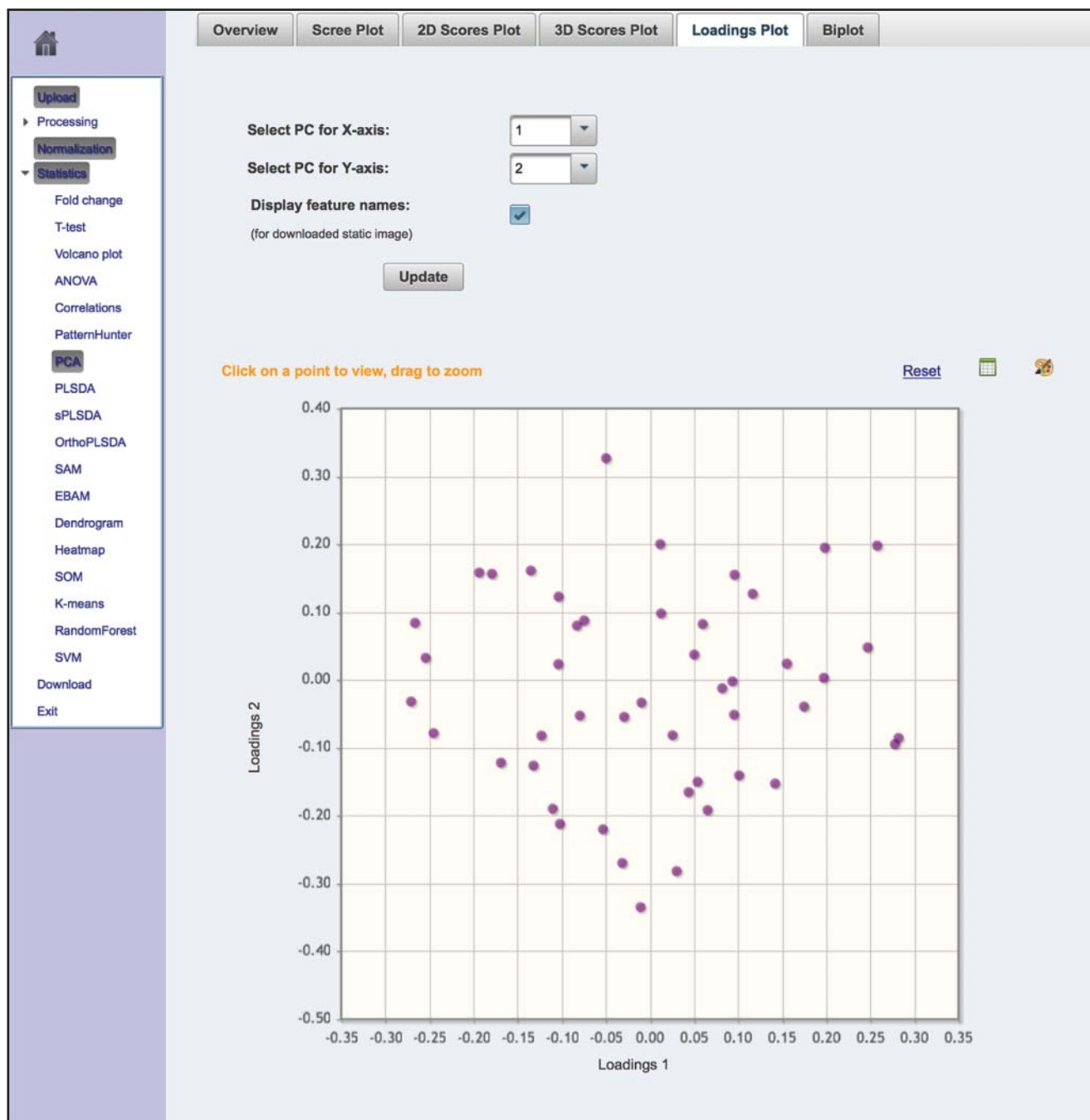


Figure 14.19 The principal component analysis (PCA) “loadings” plot, showing the most informative or statistically significant metabolites that drive the separation seen in the “scores” plot in Figure 14.18. See text for details.

compounds have been identified and quantified using targeted metabolomic methods. If the compounds are not identified prior to analysis (as in untargeted metabolomics), then the loadings plot can be used to narrow down the list of features or peaks to just a few important ones that need to be identified.

Partial Least Squares Discriminant Analysis

PCA is one of many multivariate statistical approaches that can be used to identify important metabolites or spectral features. Another type of multivariate statistical method that can be

used for this purpose is known as supervised classification. Supervised classifiers require that information about the class identities must be provided in advance of running the analysis. In other words, prior knowledge about which samples belong to the “cases” and which samples belong to the “controls” is used to label each of the samples. Examples of supervised classifiers include Soft Independent Modeling of Class Analogy (SIMCA), PLS-DA, and Orthogonal Projection of Latent Structures – Discriminant Analysis (OPLS-DA). All of these techniques can be used to help convert extensive NMR, LC-MS/MS, and GC-MS metabolite lists (for targeted metabolomics) or their corresponding spectral features (for untargeted metabolomics) into much shorter lists of highly significant metabolites and/or features.

PLS-DA is often used when PCA techniques do not generate the clusters that were expected. In particular, PLS-DA can be used to enhance the separation between data points in a PCA scores plot by essentially rotating the PCA components such that a maximum separation among classes is obtained. This separation enhancement allows one to better understand which variables are most responsible for separating the observed (or apparent) classes. The basic principles behind PLS-DA are similar to those of PCA. However, in PLS-DA a second piece of information is used – the labeled set of class identities. This extra information is used to optimize the PCs and train the clustering process.

Formally, PLS-DA is a categorical extension of PCA that takes advantage of a priori class information to attempt to maximize the covariance between the test variables and the training variable(s). Continuing with the MetaboAnalyst example described earlier, it is possible to easily generate a PLS-DA plot after generating and analyzing the PCA plots. To do so, go back to the MetaboAnalyst analysis page and click the *PLSDA* hyperlink on the page. Note that there are several PLS-DA options (regular PLS-DA, sparse PLS-DA, and orthogonal PLS-DA), so make sure to choose the regular version. Wait for 5–10 seconds in order for MetaboAnalyst to finish its default analysis. The results are then presented in a multi-panel page, with the pairwise score plots of the first five components shown as a default. Click the *2D Scores Plot* tab at the top of the page to view the scores plot between the first two PLS components (Figure 14.20). A much better separation is obtained with PLS-DA than with PCA (Figure 14.18). From here, it is possible to perform additional assessments of the quality of the PLS-DA separation using permutation testing or by plotting the R^2/Q^2 data that are discussed below. Care must be taken in using PLS-DA methods because these classification techniques can be over-trained. That is, PLS-DA can create convincing clusters or classes that are not generalizable outside of the data they are trained on (i.e. they over-fit the data). The best way of avoiding these problems is to test the resulting model on an independent dataset. However, such independent data are not always available, so a practical way to address the over-fitting problem is to use N -fold cross-validation methods or permutation (random relabeling) approaches to estimate how generalizable the data clusters derived by PLS-DA are. A number of freely available metabolomic software packages and web servers such as MetaboAnalyst and Galaxy-M are able to perform these tests. Another way of quantitatively assessing a PLS-DA model is to report R^2 and/or Q^2 values. Metabolomic web servers and software packages such as MetaboAnalyst or SIMCA typically report both R^2 and Q^2 . An example of an R^2/Q^2 plot generated by MetaboAnalyst is shown in Figure 14.21. R^2 is the correlation index and refers to the goodness of fit or the explained variation. On the other hand, Q^2 refers to the predicted variation or quality of prediction. R^2 is a quantitative measure (with a maximum value of 1 and a minimum value of 0) that indicates how well the PLS-DA model is able to mathematically reproduce the data in the dataset. A poorly fit model will have an R^2 of 0.2 or 0.3, while a nicely fit model will have an R^2 of 0.7 or 0.8. To guard against over-fitting, Q^2 is commonly determined (which also has a maximum value of 1 and a minimum of 0). Q^2 is usually estimated by cross-validation or permutation testing to assess the predictive ability of the model relative to the number of components used in the PLS-DA model. In practice, Q^2 typically tracks very closely to R^2 . However, if the PLS-DA model becomes over-fit, Q^2 reaches a maximum value and then begins to fall. Generally, a $Q^2 > 0.5$ is considered good while a Q^2 of 0.9 is outstanding.



Figure 14.20 The partial least squares discriminant analysis (PLS-DA) plot showing the separation achieved from analyzing the ruminal fluid from four different groups of cows fed four different diets. See text for details.

From a PLS-DA analysis, it is possible to use the resulting data to generate another kind of plot called the variable importance in projection (VIP) graph. An example of a VIP graph is shown in Figure 14.22. The data used to create this VIP graph are the same as those used in the PCA and PLS-DA examples given at the beginning of this section. The significance of each metabolite is plotted numerically along the X-axis (the VIP score or regression coefficient), while the metabolite name and its ranking (in importance) is shown on the Y-axis. Generally, a VIP score greater than 1.0 is significant, while a VIP score greater than 2.0 is very significant. From this graph, we can see that the same significant metabolites identified via the PCA loadings plot are again identified via the VIP plot, with aspartate, isobutyrate,

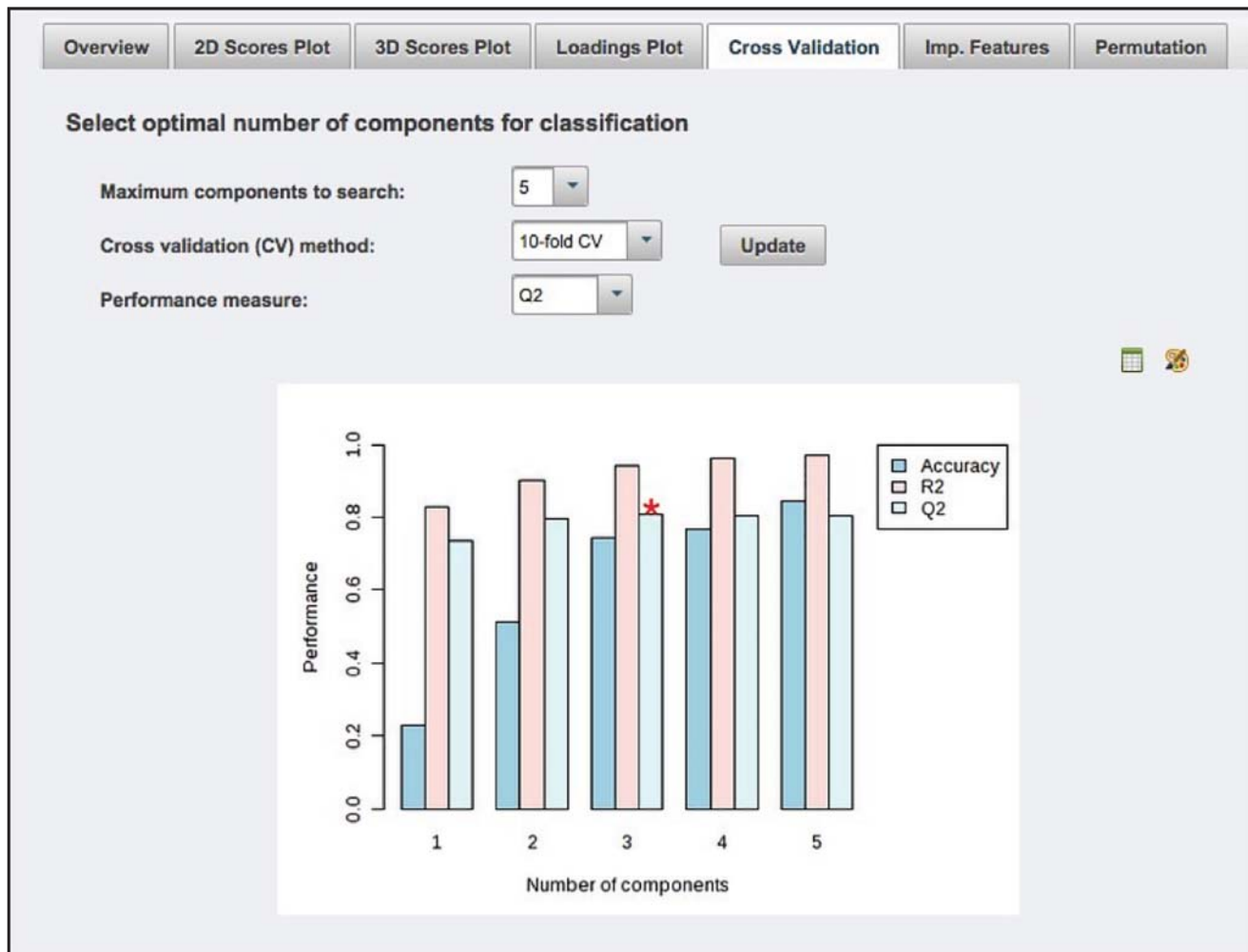


Figure 14.21 An example of an R^2/Q^2 plot generated by MetaboAnalyst using the same data for the bovine feeding experiments described in the MetaboAnalyst example in the text.

3-phenylpyruvate, endotoxin, glucose, and methylamine being at the top of the VIP plot and, therefore, being the most important.

There are a wide variety of other classification methods and metabolite/feature selection procedures that use statistical procedures (such as OPLS-DA) or machine learning protocols (such as support vector machines, random forest techniques, and artificial neural networks) to help identify significant metabolites/features from starting lists of metabolites or spectral features. These same techniques can be used in conjunction with logistic or linear regression techniques to identify important metabolite biomarkers. Many of these kinds of advanced analyses are easily accessible through tools such as MetaboAnalyst. A more detailed review of MetaboAnalyst and how it can be used to assist with metabolomic data analysis, biomarker detection, and data reduction is available in Xia and Wishart (2016).

Bioinformatics for Metabolite Interpretation

Identifying significant metabolites allows one to eliminate the noise of inconsequential or irrelevant metabolites in a metabolomic study. Once a relatively small set of significant metabolites has been identified, it becomes easier to interpret the metabolomic data. Metabolite interpretation often involves determining whether the identified metabolites belong to a single pathway

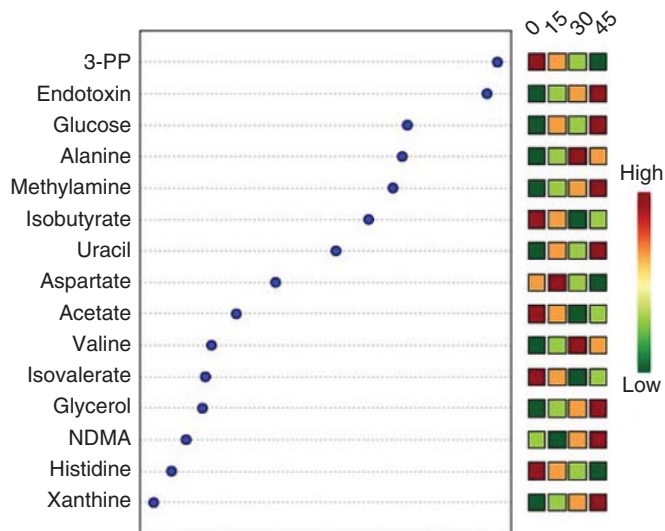


Figure 14.22 A variable importance in projection plot showing which metabolites are most important in driving the separation for the data for the bovine feeding experiments described in the MetaboAnalyst example in the text. This plot was generated using MetaboAnalyst.

or a smaller set of related pathways. In many cases, this requires searching or reading carefully through various online metabolomic databases such as the HMDB, YMDB, or others. It may also involve conducting literature reviews to see what is known about each of these metabolites and how they may act to cause the observed phenotypes.

Nearly all of the major pathway databases – including the KEGG, Reactome, Cyc databases, WikiPathways, and SMPDB – permit users to load metabolite data and to generate plots highlighting the location of key metabolites in a given pathway. The type of organism being studied and the type of pathway that needs to be illustrated often dictate the appropriate choice of database. Most metabolite/metabolism databases (such as KEGG, the Cyc databases, WikiPathways, and Reactome) only contain anabolic or catabolic pathways associated with endogenous metabolite synthesis or breakdown. Almost no information is provided on metabolite signaling pathways (such as the signaling effects of arachidonic acid), disease metabolic pathways (such as the Warburg effect), metabolic diseases (such as phenylketonuria), or drug action pathways (showing how aspirin works). As a result, many metabolomic pathway analyses are limited to interpreting complex metabolite data in only the simplest of terms (i.e. catabolic or anabolic reactions). An important exception to this is the SMPDB. This resource contains more than 700 metabolite pathways including hundreds of anabolic/catabolic pathways, dozens of signaling pathways, as well as hundreds of disease and drug pathways. Currently, the SMPDB is the only open access database that covers such a broad diversity of pathways – especially for small molecules. However, the SMPDB only contains pathways associated with humans (and other higher mammals), so it is not particularly useful for researchers doing metabolomic studies in plants, microbes, parasites, fish, or insects.

While the grouping of metabolites into known metabolic pathways can provide some important insight into their biological roles, it is also important to consider their context within specific pathways. In this regard, a new kind of software tool called MetPA (Xia and Wishart 2010a) has been developed to further facilitate pathway analysis. MetPA is a freely accessible web server that combines several pathway enrichment analysis procedures with the analysis of pathway topological characteristics to help identify the most relevant metabolic pathways involved in a given metabolomic study. Like a number of metabolomic web server applications, MetPA uses simple point-and-click operations to allow users to perform complex statistical analyses. MetPA supports three types of analyses: pathway enrichment analysis, pathway topological analysis, and pathway impact analysis. (See Chapter 13

for more information about pathway enrichment analysis.) Pathway enrichment analysis can be done using either over-representation analysis or via metabolite set enrichment analysis (MSEA) using Fisher's exact test, the hypergeometric test, and GlobalAncova (Xia and Wishart 2010a). Pathway topological analysis is based on the centrality measures of a metabolite in a given metabolic network. Centrality is a quantitative measure of the position of a metabolite relative to the other metabolites in a pathway and can be used to estimate a metabolite's relative importance or role in a pathway or network diagram. Since metabolic networks or pathways are directed graphs, MetPA uses relative "betweenness" centrality and "out-degree" centrality measures to calculate the relative importance of a metabolite. This means that metabolites located on the periphery of a pathway or those that are involved in side reactions have little consequence and are not particularly "central." On the other hand, metabolites that are in pathway bottlenecks or those that serve as hubs or precursors for many reactions are often more central. By calculating the topological importance of different metabolites in a given pathway, as well as the enrichment of certain metabolites in a pathway, it is possible to calculate a pathway impact score. Formally, the pathway impact score is the sum of the importance measures of the matched metabolites normalized by the sum of the importance measures of all metabolites in each pathway. By plotting the pathway impact score versus the number of significant metabolites appearing in that pathway (as a $-\log(P)$ value using metabolite set enrichment criteria), it is possible to generate the plot shown in Figure 14.23.

This plot illustrates the most important pathways detected from a set of approximately 30 significantly altered metabolites in a given metabolomic experiment. The pathway impact score is plotted on the X-axis and the significance of the pathway (as measured by its level of enrichment by the highly significant metabolites) is plotted on the Y-axis. The size of the circles represents the number of metabolites in the particular pathway and the color of the

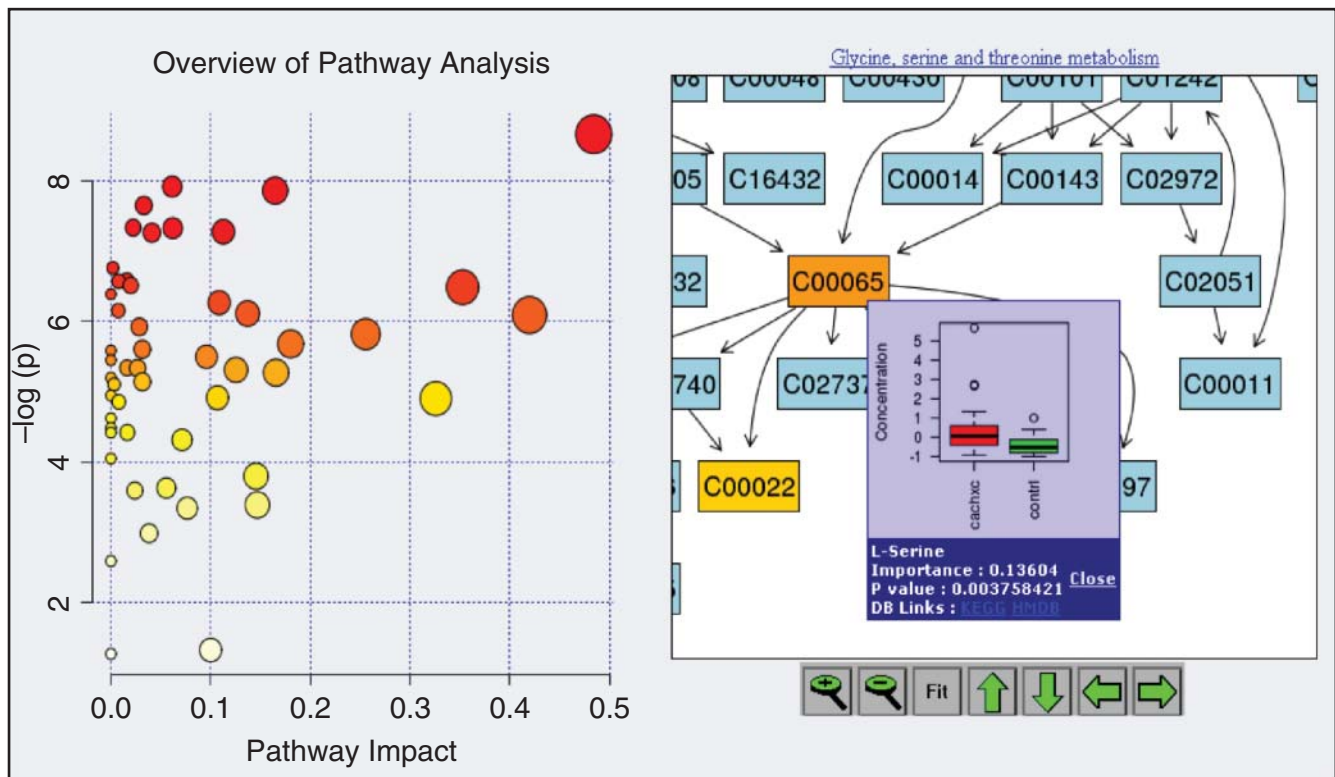


Figure 14.23 A pathway impact plot showing the importance of different pathways where significant metabolites were found from the bovine feeding experiments described in the MetaboAnalyst example in the text. This plot was generated using the pathway analysis module in MetaboAnalyst.

circle indicates its overall significance, with red being most significant and pale yellow or white being least significant. By clicking on the circles, it is possible to see a zoomable view of the pathway that shows the pathway name, the pathway components, and their topological relationships. Each detected metabolite is also clickable, allowing for a box-and-whisker plot that illustrates the metabolite concentrations and range between the case and control samples to be generated. MetPA has recently been integrated into MetaboAnalyst and it now has a database of nearly 900 different pathways collected from 21 different model organisms. Several other pathway mapping or metabolite networking approaches have emerged recently. These include MetaMapp (Barupal et al. 2012) and MetaMapR (Grapov et al. 2015). MetaMapp combines biochemical reactions from KEGG with chemical similarity and mass spectral similarity scores. This approach allows one to construct extended metabolite networks that map both identified and unidentified metabolites to potential pathways and network clusters. MetaMapR takes this concept one step further, as it allows users to calculate both structural and mass spectral similarity directly within the program, while at the same time supporting interactive network visualization.

In addition to pathway and/or network analysis, there are a number of other approaches that can be used to interpret, visualize, or explore metabolomic data. One particularly useful approach involves using MSEA (Xia and Wishart 2010b). MSEA is a form of functional enrichment analysis similar to gene set enrichment analysis (GSEA). For metabolite set enrichment to be effective it is necessary for the software to have either a comprehensive database of metabolic pathways, a database of healthy/diseased metabolite levels, or a database of associations between metabolites and single nucleotide polymorphisms (SNPs) or metabolites and gene expression levels. Ideally, a good MSEA system should have all of these databases and support all of these functional analyses. Another approach to interpreting metabolomic data is to combine them with gene expression or protein expression data (Xia et al. 2013). There are a number of emerging approaches that support this kind of integration including MetScape (Karnovsky et al. 2012). MetScape is a plug-in for a widely used open source network analysis and visualization tool called Cytoscape (see Chapter 13). MetScape supports the interactive, network-based exploration and visualization of both metabolite and gene expression data by integrating both the KEGG and Edinburgh Human Metabolic Network databases. MetScape allows users to identify enriched pathways from gene or metabolite expression profiling data, build and analyze gene or metabolite networks, and interactively visualize changes in gene or metabolite data. Another integrated “omics” approach that offers similar capabilities is called Integrated Metabolomic and Expression Analysis (INMEX; Xia et al. 2013). This web-based tool is now available through MetaboAnalyst. Like MetScape, INMEX makes use of the KEGG pathway database, as well as a number of pathways from the SMPDB.

Another bioinformatic technique that can also be used to interpret metabolomic data involves metabolic simulations and metabolic flux balance analysis (Lewis et al. 2012). These techniques typically require a detailed reconstruction of the entire organism’s metabolic pathways that consider mass and charge balance, metabolite compartmentalization, and known or estimated metabolite concentrations. They also require detailed knowledge of the genes, proteins, and cofactors required for all of the enzymatic and metabolic transport reactions. Metabolic reconstructions and metabolic simulations have been described for a number of organisms including *E. coli*, yeast, *Caenorhabditis elegans*, *Arabidopsis*, and even humans (Ruppin et al. 2010; Lewis et al. 2012; Swainston et al. 2016). These metabolic reconstructions have been used to predict the consequences of mutations in metabolic pathways, to rationalize the appearance of certain metabolites in certain physiological or disease-associated conditions, and to help predict the presence of previously undetected or unexpected compounds. These remarkable simulations represent the pinnacle of what can be achieved through combining high-level bioinformatics with high-level metabolomics. They also serve as superb examples of how metabolomics can serve as a foundational tool to allow bioinformaticians to conduct advanced research into systems biology.

Summary

The field of metabolomics involves a unique blend of basic biology and analytical chemistry, along with a generous helping of bioinformatics, cheminformatics, and statistics. The fact that metabolomic approaches have led to a number of important biomedical discoveries (Wang et al. 2011a, b) and are opening the door to many more (Wishart 2016) has made these approaches increasingly popular among life science researchers. Indeed, the field of metabolomics has grown considerably in size, scope, and sophistication over the past decade. As a result, a detailed description of the many bioinformatic/cheminformatic tools, resources, and techniques that have been developed for metabolomics could easily fill several books. This chapter is only intended to serve as an easily accessible gateway so that individuals who are interested in pursuing metabolomics and using bioinformatic or cheminformatic tools for metabolomics can better appreciate what is available, what is possible, and what still needs to be done.

Internet Resources

ACD/ChemSketch	www.acdlabs.com/resources/freeware/chemsketch
Avogadro	avogadro.cc
BATMAN	batman.r-forge.r-project.org
Bayesil	bayesil.ca
BioMagResBank	www.bmrb.wisc.edu/metabolomics/
CFM-ID	cfmid.wishartlab.com
Chemical Entities of Biological Interest (ChEBI)	www.ebi.ac.uk/chebi
ChemSpider	www.chemspider.com
<i>E. coli</i> Metabolome Database (ECMDB)	ecmdb.ca
Galaxy-M	github.com/Viant-Metabolomics/Galaxy-M
GolmDB	gmd.mpimp-golm.mpg.de
Human Metabolome Database (HMDB)	www.hmdb.ca
HTML5 Molecular Editor	www.molsoft.com/moledit.html
JChemPaint	jchempaint.github.io
JDXview	merian.pch.univie.ac.at/~nhaider/cheminf/jdxview.html
JSME	peter-ertl.com/jsme
JSmol	sourceforge.net/projects/jsmol
JSpectraViewer	github.com/sciguy/jspectra_viewer
JSpecView	sourceforge.net/projects/jspecview
Kyoto Encyclopedia of Genes and Genomes (KEGG)	www.genome.jp/kegg
KNApSAcK	kanaya.naist.jp/KNApSAcK/KNApSAcK.php
KnowItAll Academic	www.bio-rad.com
LIPID MAPS	www.lipidmaps.org
MarvinSketch	www.chemaxon.com/products/marvin/marvinsketch
MeltDB 2	meltdb.cebitec.uni-bielefeld.de/cgi-bin/login.cgi
Metabolomics Workbench	www.metabolomicsworkbench.org
MetaboAnalyst	www.metaboanalyst.ca
MetaboLights	www.ebi.ac.uk/metabolights
MetaCyc	metacyc.org
Meta-P	metap.helmholtz-muenchen.de/metap2
METLIN	metlin.scripps.edu/landing_page.php?pgcontent=mainPage

MassBank of North America (MoNA)	mona.fiehnlab.ucdavis.edu
MS-DIAL	prime.psc.riken.jp/Metabolomics_Software/MS-DIAL
MZmine 2	mzmine.github.io
NMRShiftDB	nmrshiftdb.nmr.uni-koeln.de
OPSIN	opsin.ch.cam.ac.uk
PubChem	pubchem.ncbi.nlm.nih.gov
R Programming Language	www.r-project.org
Reactome	www.reactome.org
Small Molecule Pathway Database (SMPDB)	smpdb.ca
Toxic Exposome Database (T3DB)	www.t3db.ca
WikiPathways	www.wikipathways.org/index.php/WikiPathways
XCMS	xcmsonline.scripps.edu
XDrawChem	www.woodsidelabs.com/chemistry/xdrawchem.php
Yeast Metabolome Database (YMDB)	www.ymdb.ca

Further Reading

- Dunn, W.B., Bailey, N.J., and Johnson, H.E. (2005). Measuring the metabolome: current analytical technologies. *Analyst* 130: 606–625. A nice review of the different technologies used in metabolomics. While the paper is a little old, the explanations are insightful and easy to understand. Papers like this never go stale.
- Kind, T. and Fiehn, O. (2010). Advances in structure elucidation of small molecules using mass spectrometry. *Bioanal. Rev.* 2: 23–60. A very comprehensive review of how mass spectrometry can and should be used to characterize metabolites. Many topics are covered in exquisite detail. The authors are highly respected mass spectroscopists and pioneered many of the techniques and ideas used in modern metabolomics.
- Wishart, D.S. (2016). Emerging applications of metabolomics in drug discovery and precision medicine. *Nat. Rev. Drug Discov.* 15: 473–484. An introduction to how metabolomics can be (and is being) used in medical applications. This highlights some of the more important and interesting biomedical discoveries to emerge from metabolomics over the past 10 years. It also looks ahead to where metabolomics will likely be going.
- Xia, J. and Wishart, D.S. (2016). Using MetaboAnalyst 3.0 for comprehensive metabolomics data analysis. *Curr. Protoc. Bioinf.* 55: 14.10.1–14.10.93. A very detailed, step-by-step description (with plenty of screenshots) describing all the tools, tips and tricks in MetaboAnalyst. This is a must-read for anyone wishing to work in the field of metabolomics and with MetaboAnalyst.

References

- Allen, F., Pon, A., Wilson, M. et al. (2014). CFM-ID: a web server for annotation, spectrum prediction and metabolite identification from tandem mass spectra. *Nucleic Acids Res.* 42 (Web Server issue): W94–W99.
- Allen, F., Greiner, R., and Wishart, D.S. (2015). Competitive fragmentation modeling of ESI-MS/MS spectra for putative metabolite identification. *Metabolomics* 11: 98–110.
- Allen, F., Pon, A., Greiner, R., and Wishart, D. (2016). Computational prediction of electron ionization mass spectra to assist in GC/MS compound identification. *Anal. Chem.* 88: 7689–7697.
- Barupal, D.K., Haldiya, P.K., Wohlgemuth, G. et al. (2012). MetaMapp: mapping and visualizing metabolomic data by integrating information from biochemical pathways and chemical and mass spectral similarity. *BMC Bioinf.* 13: 99.

- Bassini, A. and Cameron, L.C. (2014). Sportomics: building a new concept in metabolic studies and exercise science. *Biochem. Biophys. Res. Commun.* 445: 708–716.
- Bienfait, B. and Ertl, P. (2013). JSME: a free molecule editor in JavaScript. *J. Cheminform.* 5: 24.
- Böcker, S., Letzel, M.C., Lipták, Z., and Pervukhin, A. (2009). SIRIUS: decomposing isotope patterns for metabolite identification. *Bioinformatics* 25: 218–224.
- Brown, S.A. (2016). Circadian metabolism: from mechanisms to metabolomics and medicine. *Trends Endocrinol. Metab.* 27: 415–426.
- Croft, D., O’Kelly, G., Wu, G. et al. (2011). Reactome: a database of reactions, pathways and biological processes. *Nucleic Acids Res.* 39 (Database issue): D691–D697.
- da Silva, R.R., Dorrestein, P.C., and Quinn, R.A. (2015). Illuminating the dark matter in metabolomics. *Proc. Natl. Acad. Sci. U.S.A.* 112: 12549–12550.
- Dalby, A., Nourse, J.G., Hounshell, W.D. et al. (1992). Description of several chemical structure file formats used by computer programs developed at Molecular Design Limited. *J. Chem. Inf. Comput. Sci.* 32: 244–255.
- Davidson, R.L., Weber, R.J., Liu, H. et al. (2016). Galaxy-M: a Galaxy workflow for processing and analyzing direct infusion and liquid chromatography mass spectrometry-based metabolomics data. *GigaScience* 5: 10.
- Demir, E., Cary, M.P., Paley, S. et al. (2010). The BioPAX community standard for pathway data sharing. *Nat. Biotechnol.* 28: 935–942.
- Deutsch, E.W. (2008). mzML: a single, unifying data format for mass spectrometer output. *Proteomics* 14: 2776–2777.
- Dunn, W.B., Bailey, N.J., and Johnson, H.E. (2005). Measuring the metabolome: current analytical technologies. *Analyst* 130: 606–625.
- Durant, J.L., Leland, B.A., Henry, D.R., and Nourse, J.G. (2002). Reoptimization of MDL keys for use in drug discovery. *J. Chem. Inf. Comput. Sci.* 42: 1273–1280.
- Ertl, P. (2010). Molecular structure input on the web. *J. Cheminform.* 2: 1.
- Fahy, E., Sud, M., Cotter, D., and Subramaniam, S. (2007). LIPID MAPS online tools for lipid research. *Nucleic Acids Res.* 35 (Web Server issue): W606–W612.
- Fiehn, O. (2002). Metabolomics – the link between genotypes and phenotypes. *Plant Mol. Biol.* 48: 155–171.
- Grapov, D., Wanichthanarak, K., and Fiehn, O. (2015). MetaMapR: pathway independent metabolomic network analysis incorporating unknowns. *Bioinformatics* 31: 2757–2760.
- Guo, A.C., Jewison, T., Wilson, M. et al. (2013). ECMDB: the *E. coli* Metabolome Database. *Nucleic Acids Res.* 41 (Database issue): D625–D630.
- Hanson, R.M., Prilusky, J., Renjian, Z. et al. (2013). JSmol and the next-generation web-based representation of 3D molecular structure as applied to Proteopedia. *Isr. J. Chem.* 53: 207–216.
- Hanwell, M.D., Curtis, D.E., Lonie, D.C. et al. (2012). Avogadro: an advanced semantic chemical editor, visualization, and analysis platform. *J. Cheminform.* 4: 17.
- Hao, J., Liebeke, M., Astle, W. et al. (2014). Bayesian deconvolution and quantification of metabolites in complex 1D NMR spectra using BATMAN. *Nat. Protoc.* 9: 1416–1427.
- Hastings, J., de Matos, P., Dekker, A. et al. (2013). The ChEBI reference database and ontology for biologically relevant chemistry: enhancements for 2013. *Nucleic Acids Res.* 41 (Database issue): D456–D463.
- Haug, K., Salek, R.M., Conesa, P. et al. (2013). MetaboLights – an open-access general-purpose repository for metabolomics studies and associated meta-data. *Nucleic Acids Res.* 41 (Database issue): D781–D786.
- Heller, S.R., McNaught, A., Pletnev, I. et al. (2015). InChI, the IUPAC international chemical identifier. *J. Cheminform.* 7: 23.
- Holmes, E., Wilson, I.D., and Nicholson, J.K. (2008). Metabolic phenotyping in health and disease. *Cell* 134: 714–717.
- Hucka, M., Finney, A., Sauro, H.M. et al. (2003). The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* 19: 524–531.

- Jewison, T., Knox, C., Neveu, V. et al. (2012). YMDB: the yeast metabolome database. *Nucleic Acids Res.* 40 (Database issue): D815–D820.
- Jewison, T., Su, Y., Disfany, F.M. et al. (2014). SMPDB 2.0: big improvements to the small molecule pathway database. *Nucleic Acids Res.* 42 (Database issue): D478–D484.
- Kanehisa, M., Goto, S., Sato, Y. et al. (2014). Data, information, knowledge and principle: back to metabolism in KEGG. *Nucleic Acids Res.* 42 (Database issue): D199–D205.
- Karnovsky, A., Weymouth, T., Hull, T. et al. (2012). Metscape 2 bioinformatics tool for the analysis and visualization of metabolomics and gene expression data. *Bioinformatics* 28: 373–380.
- Karp, P.D., Riley, M., Saier, M. et al. (2000). The EcoCyc and MetaCyc databases. *Nucleic Acids Res.* 28: 56–59.
- Kastenmüller, G., Römisch-Margl, W., Wägele, B. et al. (2011). metaP-server: a web-based metabolomics data analysis tool. *J. Biomed. Biotechnol.* 2011 <https://doi.org/10.1155/2011/839862>.
- Katajamaa, M., Miettinen, J., and Oresic, M. (2006). MZmine: toolbox for processing and visualization of mass spectrometry based molecular profile data. *Bioinformatics* 22: 634–636.
- Kelder, T., van Iersel, M.P., Hanspers, K. et al. (2012). WikiPathways: building research communities on biological pathways. *Nucleic Acids Res.* 40 (Database issue): D1301–D1307.
- Kessler, N., Neuweger, H., Bonte, A. et al. (2013). MeltDB 2.0-advances of the metabolomics software system. *Bioinformatics* 29: 2452–2459.
- Kim, S., Kim, J., Yun, E.J., and Kim, K.H. (2016). Food metabolomics: from farm to human. *Curr. Opin. Biotechnol.* 37: 16–23.
- Kind, T. and Fiehn, O. (2007). Seven Golden Rules for heuristic filtering of molecular formulas obtained by accurate mass spectrometry. *BMC Bioinf.* 8: 105.
- Kind, T. and Fiehn, O. (2010). Advances in structure elucidation of small molecules using mass spectrometry. *Bioanal. Rev.* 2: 23–60.
- Kind, T., Tsugawa, H., Cajka, T. et al. (2017). Identification of small molecules using accurate mass MS/MS search. *Mass Spectrom. Rev.* <https://doi.org/10.1002/mas.21535>.
- Kopka, J., Schauer, N., Krueger, S. et al. (2005). GMD@CSB.DB: the Golm metabolome database. *Bioinformatics* 21: 1635–1638.
- Krause, S., Willighagen, E., and Steinbeck, C. (2000). JChemPaint – using the collaborative forces of the internet to develop a free editor for 2D chemical structures. *Molecules* 5: 93–98.
- Kuhn, S., Helmus, T., Lancashire, R.J. et al. (2007). Chemical markup, XML, and the World Wide Web. 7. CMLSpect, an XML vocabulary for spectral data. *J. Chem. Inf. Model.* 47: 2015–2034.
- Lancashire, R.J. (2007). The JSpecView Project: an Open Source Java viewer and converter for JCAMP-DX, and XML spectral data files. *Chem. Cent. J.* 1: 31.
- Levy, P.A. (2010). An overview of newborn screening. *J. Dev. Behav. Pediatr.* 31: 622–631.
- Lewis, N.E., Nagarajan, H., and Palsson, B.O. (2012). Constraining the metabolic genotype-phenotype relationship using a phylogeny of in silico methods. *Nat. Rev. Microbiol.* 10: 291–305.
- Lu, H., Liang, Y., Dunn, W.B. et al. (2008). Comparative evaluation of software for deconvolution of metabolomics data based on GC-TOF-MS. *Trends Anal. Chem.* 27: 215–227.
- Markley, J.L., Ulrich, E.L., Berman, H.M. et al. (2008). BioMagResBank (BMRB) as a partner in the Worldwide Protein Data Bank (wwPDB): new policies affecting biomolecular NMR depositions. *J. Biomol. NMR.* 40: 153–155.
- McDonald, R.S. and Wilks, P.A. (1988). JCAMP-DX: a standard form for exchange of infrared spectra in computer-readable form. *Appl. Spectrosc.* 42: 151–162.
- Melamud, E., Vastag, L., and Rabinowitz, J.D. (2010). Metabolomic analysis and visualization engine for LC-MS data. *Anal. Chem.* 82: 9818–9826.
- Nakamura, K., Shimura, N., Otabe, Y. et al. (2013). KNApSACk-3D: a three-dimensional structure database of plant metabolites. *Plant Cell Physiol.* 54: e4.
- Naz, S., Moreira dos Santos, D.C., Garcia, A., and Barbas, C. (2014). Analytical protocols based on LC-MS, GC-MS and CE-MS for nontargeted metabolomics of biological tissues. *Bioanalysis* 6: 1657–1677.

- Niedermeyer, T.H. (2016). Annotating and interpreting linear and cyclic peptide tandem mass spectra. *Methods Mol. Biol.* 1401: 199–207.
- O’Boyle, N.M., Banck, M., James, C.A. et al. (2011). Open Babel: an open chemical toolbox. *J. Cheminform.* 3: 33.
- Psychogios, N., Hau, D.D., Peng, J. et al. (2011). The human serum metabolome. *PLoS One* 6 (2): e16957.
- Ravanbakhsh, S., Liu, P., Bjorndahl, T.C. et al. (2015). Accurate, fully-automated NMR spectral profiling for metabolomics. *PLoS One* 10: e0124219.
- Ruppin, E., Papin, J.A., de Figueiredo, L.F., and Schuster, S. (2010). Metabolic reconstruction, constraint-based analysis and game theory to probe genome-scale metabolic networks. *Curr. Opin. Biotechnol.* 21: 502–510.
- Schober, D., Jacob, D., Wilson, M. et al. (2018). nmrML: an open standard for the description, storage and exchange on NMR data. *Anal. Chem.* 90: 649–656.
- Smith, C.A., Want, E.J., O’Maille, G. et al. (2006). XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. *Anal. Chem.* 78: 779–787.
- Stein, S.E. (1999). An integrated method for spectrum extraction and compound identification from gas chromatography/mass spectrometry data. *J. Am. Soc. Mass Spectrom.* 10: 770–781.
- Steinbeck, C. and Kuhn, S. (2004). NMRShiftDB – compound identification and structure elucidation support through a free community-built web database. *Phytochemistry* 65: 2711–2717.
- Steinbeck, C., Hoppe, C., Kuhn, S. et al. (2006). Recent developments of the chemistry development kit (CDK) an open-source java library for chemo- and bioinformatics. *Curr. Pharm. Des.* 12: 2111–2120.
- Sud, M., Fahy, E., Cotter, D. et al. (2016). Metabolomics Workbench: an international repository for metabolomics data and metadata, metabolite standards, protocols, tutorials and training, and analysis tools. *Nucleic Acids Res.* 44 (D1): D463–D470.
- Sumner, L.W., Amberg, A., Barrett, D. et al. (2007). Proposed minimum reporting standards for chemical analysis. *Metabolomics* 3: 211–221.
- Swainston, N., Smallbone, K., Hefzi, H. et al. (2016). Recon 2.2: from reconstruction to model of human metabolism. *Metabolomics* 12: 109.
- Tautenhahn, R., Cho, K., Uritboonthai, W. et al. (2012). An accelerated workflow for untargeted metabolomics using the METLIN database. *Nat. Biotechnol.* 30: 826–828.
- Tsugawa, H., Cajka, T., Kind, T. et al. (2015). MS-DIAL: data-independent MS/MS deconvolution for comprehensive metabolome analysis. *Nat. Methods* 12: 523–526.
- van Iersel, M.P., Villéger, A.C., Czauderna, T. et al. (2012). Software support for SBGN maps: SBGN-ML and LibSBGN. *Bioinformatics* 28: 2016–2021.
- Viant, M.R. (2008). Recent developments in environmental metabolomics. *Mol. Biosyst.* 4: 980–986.
- Wang, T.J., Larson, M.G., Vasan, R.S. et al. (2011a). Metabolite profiles and the risk of developing diabetes. *Nat. Med.* 17: 448–453.
- Wang, Z., Klipfell, E., Bennett, B.J. et al. (2011b). Gut flora metabolism of phosphatidylcholine promotes cardiovascular disease. *Nature* 472: 57–63.
- Weininger, D. (1988). SMILES 1. Introduction and encoding rules. *J. Chem. Inf. Comput. Sci.* 28: 31–38.
- Westbrook, J.D. and Fitzgerald, P.M. (2003). The PDB format, mmCIF, and other data formats. *Methods Biochem. Anal.* 44: 161–179.
- Wheeler, D.L., Barrett, T., Benson, D.A. et al. (2006). Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.* 34 (Database issue): D173–D180.
- Wild, C.P. (2005). Complementing the genome with an “exposome”: the outstanding challenge of environmental exposure measurement in molecular epidemiology. *Cancer Epidemiol. Biomarkers Prev.* 14: 1847–1850.
- Williams, A.J. (2008). Public chemical compound databases. *Curr. Opin. Drug Discov. Devel.* 11: 393–404.

- Wishart, D.S. (2005). Metabolomics: the principles and potential applications to transplantation. *Am. J. Transplant.* 5: 2814–2820.
- Wishart, D.S. (2008). Quantitative metabolomics using NMR. *Trends Anal. Chem.* 27: 228–237.
- Wishart, D.S. (2011). Advances in metabolite identification. *Bioanalysis* 3: 1769–1782.
- Wishart, D.S. (2016). Emerging applications of metabolomics in drug discovery and precision medicine. *Nat. Rev. Drug Discov.* 15: 473–484.
- Wishart, D.S., Tzur, D., Knox, C. et al. (2007). HMDB: the human metabolome database. *Nucleic Acids Res.* 35 (Database issue): D521–D526.
- Wishart, D.S., Arndt, D., Pon, A. et al. (2015). T3DB: the toxic exposome database. *Nucleic Acids Res.* 43 (Database issue): D928–D934.
- Worley, B. and Powers, R. (2014). MVAPACK: a complete data handling package for NMR metabolomics. *ACS Chem. Biol.* 9: 1138–1144.
- Xia, J. and Wishart, D.S. (2010a). MetPA: a web-based metabolomics tool for pathway analysis and visualization. *Bioinformatics* 26: 2342–2344.
- Xia, J. and Wishart, D.S. (2010b). MSEA: a web-based tool to identify biologically meaningful patterns in quantitative metabolomic data. *Nucleic Acids Res.* 38 (Web Server issue): W71–W77.
- Xia, J. and Wishart, D.S. (2016). Using MetaboAnalyst 3.0 for comprehensive metabolomics data analysis. *Curr. Protoc. Bioinf.* 55: 14.10.1–14.10.93.
- Xia, J., Fjell, C.D., Mayer, M.L. et al. (2013). INMEX – a web-based tool for integrative meta-analysis of expression data. *Nucleic Acids Res.* 41 (Web Server issue): W63–W70.
- Xia, J., Sinelnikov, I.V., Han, B., and Wishart, D.S. (2015). MetaboAnalyst 3.0 – making metabolomics more meaningful. *Nucleic Acids Res.* 43 (W1): W251–W257.

15

Population Genetics

Lynn B. Jorde and W. Scott Watkins

Introduction

Population genetics can be defined as the study of genetic variation within and between populations. Population genetics has been applied to help understand the evolution of plants, insects, fish, wildlife, livestock, and humans. The methods described in this chapter can be applied to any of these organisms, but the data and examples discussed here will focus on human populations. In the past several years, whole genome sequencing has allowed investigators an unparalleled view of all variation in the human genome and has greatly enhanced our ability to study and understand human genetic variation (Auton et al. 2015; Mallick et al. 2016). It is now possible to reconstruct detailed accounts of human demographic history and to understand many of the ways in which humans have adapted to ever-changing environments.

Evolutionary Processes and Genetic Variation

To measure and analyze genetic variation, it is important to understand the evolutionary processes that give rise to it. Four fundamental processes will be discussed: mutation, natural selection, gene flow, and genetic drift.

Mutation is the ultimate source of all genetic variation and, within species, occurs with almost clocklike regularity through time. Thus, an accurate estimate of the rate of mutation in a species can be used to estimate the dates of major events in the species history. In humans, mutation rates have been approximated by comparing differences in human DNA sequences with those of chimpanzees, our closest non-human relatives. This mutation rate, which is really a substitution rate, measures the rate at which new variants survive and accumulate in a lineage. Then, assuming a human–chimp divergence date of roughly 6 million years ago, a mutation (substitution) rate of roughly 2.5×10^{-8} per nucleotide per generation has been estimated and widely used (Nachman and Crowell 2000). More recently, human mutation rates have been estimated directly by comparing whole genome sequences in parents and offspring in families (Roach et al. 2010; Conrad et al. 2011; Moorjani et al. 2016). Somewhat surprisingly, these estimates, which average about 1.2×10^{-8} per nucleotide per generation, are roughly half of the substitution rate estimate obtained by human–chimp comparisons. This difference may reflect uncertainties about the human–chimp divergence time and the size of the ancestral human–chimp population (Campbell and Eichler 2013). In addition, direct estimates of mutation rates are influenced by factors such as parental age and the effects of somatic mutations in sampled tissue (typically blood) (Shendure and Akey 2015; Moorjani et al. 2016). Thus, some uncertainty remains regarding the most appropriate rate for evolutionary applications (Segurel et al. 2014).

Natural selection is the process through which the population frequencies of deleterious genetic variants are reduced and the frequencies of favorable variants are increased. Importantly, some variants may be advantageous in certain environments (e.g. sickle cell disease in

malarial environments) but harmful in others. For most human disease-causing genes, mutation introduces deleterious variants, while natural selection tends to eliminate them – a process known as mutation–selection balance. With the availability of genome-scale data such as single nucleotide variants (SNVs) ascertained through microarrays or whole genome sequencing, the process of natural selection can be analyzed systematically in human populations (Fu and Akey 2013). As discussed below, there is an increasing number of examples of recent natural selection in human populations.

Gene flow is the process in which genetic variation is exchanged among populations. Humans are a highly mobile species, so genetic variation often tends to change gradually across geographic space (Rosenberg et al. 2005). In general, gene flow acts as a homogenizing influence on genetic variation between populations. As with natural selection, large-scale genomic data now enable sensitive and fine-grained analysis of gene flow patterns in populations (Hellenthal et al. 2014).

Genetic drift refers to changes in genetic variation that occur through time because of finite population size. In small populations, the frequencies of genetic variants can change rapidly because only a small number of variants are being transmitted to the next generation. (To imagine this, consider a coin-tossing experiment in which only 10 coins are tossed: the frequencies of “heads” and “tails” can vary dramatically from the expected 50%; if thousands of coins are tossed, these frequencies will remain very close to the expected 50%). Genetic drift can produce high frequencies of specific diseases in populations that are (or recently were) very small. For instance, the incidence of certain rare diseases such as Ellis–van Creveld disease are highly elevated in the Old Order Amish (Strauss and Puffenberger 2009), while Tay–Sachs disease, Gaucher disease, and Niemann–Pick disease are elevated in the Ashkenazi Jewish population (Ostrer and Skorecki 2013). In general, whereas gene flow tends to homogenize populations, genetic drift tends to differentiate them.

Allele Frequencies and Population Variation

The four evolutionary factors just discussed cause changes in allele frequencies (or gene frequencies) in populations. Allele frequencies are measured as the proportion of chromosome copies that contain a given allele (i.e. in a population of 100 people, there are 200 chromosome copies, so if 50 copies of allele A are seen then the frequency of A is $50/200$ or 0.25). Allele frequencies can be estimated for all types of genetic variation, including SNVs and copy number variants (CNVs).

Allele frequencies are often displayed as a histogram of the proportion of alleles in frequency bins ranging from 0 to 0.5; this represents the allele frequency spectrum or site frequency spectrum. The allele frequency spectrum can be used to make inferences about the effects of evolutionary factors like genetic drift and natural selection. For example, newly arisen alleles are not lost to genetic drift in rapidly expanding populations, producing an excess of rare alleles (Novembre and Ramachandran 2011). High frequencies of rare alleles are typically observed in human populations, providing strong evidence that many human populations underwent major expansions in size during the past 50 000 years (Tennessen et al. 2012). Furthermore, rapid expansion of population size can limit the capacity of natural selection to remove deleterious alleles, and it is estimated that 85% of all deleterious variants in the human genome have arisen in just the past 5000–10 000 years (Fu et al. 2013).

Population allele frequencies can also be used to estimate genetic distances between pairs of populations. The simplest form of genetic distance consists of the absolute value of the difference in allele frequencies, averaged across all loci. When n populations are studied, an $n \times n$ genetic distance matrix can be formulated and analyzed with various data reduction and display methods (described below). Genetic distances can also be estimated between pairs of individuals.

One of the most commonly used measures of population genetic differentiation is known as the fixation index, or F_{ST} . It is estimated by $(H_T - H_S)/H_T$, where H_S is the heterozygosity (Box 15.1) within each population, averaged across all populations, and H_T is the total heterozygosity in all individuals, combining them as a single population. If $F_{ST} = 0$, then there is just as much variation, on average, within populations as there is across the entire sample. If $F_{ST} = 1$, then there is no variation within populations ($H_S = 0$), with all variation due to differences between populations. F_{ST} , then, is a measure of between-population variation (Holsinger and Weir 2009). F_{ST} has been studied extensively in human populations (Holsinger and Weir 2009); among major continents, it is approximately 0.10–0.15 (Witherspoon et al. 2007). Thus, most variation in humans can be found within one of the world's major continents (e.g. Asia or Africa), and only a relatively small additional amount of variation is contributed by between-continent differences. Among populations within the same continent, F_{ST} is typically smaller, varying from 0.01 to 0.05 (Auton et al. 2015; Novembre and Peter 2016). In contrast, for loci that have undergone strong natural selection in different environments, such as genes that influence skin pigmentation, F_{ST} between continents can exceed 0.90 (Lamason et al. 2005).

Box 15.1 Basic Definitions and Concepts

Heterozygosity is a measure of genetic variation. For a single nucleotide variant (SNV) with alleles A and T, an individual can be either homozygous (genotypes AA or TT) or heterozygous (AT). In a population, an SNV's heterozygosity is measured simply as the proportion of individuals who have the heterozygous (AT) genotype. Average heterozygosity is simply the mean heterozygosity across all measured SNVs.

The Hardy–Weinberg principle specifies the expected relationship between allele frequencies and genotype frequencies in a population. It is assumed that individuals mate randomly with respect to the SNV under consideration, which allows us to apply the multiplication and addition rules of probability. For the SNV just discussed, imagine that the allele frequency of A in a population is 0.60 and the frequency of T is 0.40. This means that 60% of sperm cells in the population would have allele A and 40% would have allele T. The same frequencies would apply to egg cells. Then, under random mating, the probability that a sperm cell carrying A unites with an egg cell carrying A is given by the product of the allele frequencies: 0.60×0.60 , or 0.36. This is the expected frequency of the AA genotype in the population. Similarly, the probability that a random mating event produces a TT genotype is 0.40×0.40 , or 0.16. A heterozygote can be produced either by the union of an A-bearing sperm cell and a T-bearing egg cell or by a T-bearing sperm cell and an A-bearing egg cell. Each of these events has a probability of 0.60×0.40 , or 0.24. To get the overall frequency of the heterozygous genotype, we simply add the two probabilities together to yield a frequency of 0.48. (Notice that the three genotype frequencies add to 1.00.) This heterozygosity estimate, which is predicted by the Hardy–Weinberg principle, is termed the *expected* heterozygosity. It can be compared with the *observed* heterozygosity, which is obtained simply by counting the heterozygotes, as in the previous paragraph. If the observed and expected heterozygosity values are not significantly different from one another, the SNV is said to be in Hardy–Weinberg equilibrium (HWE). Deviations from HWE can be caused by the mating of close relatives or by population stratification, in which subgroups of the population are more likely to mate among themselves. Each of these phenomena decreases the observed heterozygosity levels, relative to the expectation under HWE.

Similar logic can be used to predict haplotype frequencies from allele frequencies in populations (a haplotype, or haploid genotype, refers to the alleles on one copy of a chromosome in an individual). As an example, consider two linked loci, with alleles A,a at one locus and B,b at the other locus. If the frequencies of A and a are 0.60 and 0.40,

(Continued)

Box 15.1 (Continued)

respectively, and the frequencies of B and b are 0.70 and 0.30, respectively, the predicted frequency of the AB haplotype would be $0.60 \times 0.70 = 0.42$ (i.e. we multiply the two allele frequencies together to estimate the frequency of their co-occurrence on the same chromosome copy). Similarly, the predicted frequencies of haplotypes Ab, aB, and ab, are 0.18, 0.28, and 0.12, respectively. If the observed frequencies of these haplotypes equal the frequencies predicted by multiplying the allele frequencies together, then the two loci are said to be in linkage equilibrium, indicating that the alleles at the two loci are statistically independent of one another. However, for very closely linked loci, recombination is so rare that certain haplotypes are observed more or less frequently than predicted under linkage equilibrium (e.g. the frequency of AB might be 0.55 instead of the predicted 0.42). When the observed haplotype frequencies differ significantly from the predicted haplotype frequencies, the two loci are in linkage disequilibrium. The existence of linkage disequilibrium typically indicates that the pair of loci are located very close together on the same chromosome.

F_{ST} estimates are influenced by the choice and geographic distribution of analyzed populations. Generally, when populations are chosen to be more continuously distributed geographically, F_{ST} levels are somewhat reduced (Xing et al. 2010). F_{ST} can be unduly influenced by high levels of drift in single populations, so methods have been formulated to eliminate this bias (Patterson et al. 2012).

Display Methods

If a large number of populations (or individuals) are sampled, an $n \times n$ genetic distance matrix becomes difficult to interpret. Statistical methods are used to reduce the complexity of a genetic distance matrix to just a few important dimensions. Phylogenetic trees are sometimes used to portray relationships among human populations (see Chapter 9), but this practice is sometimes questioned because the tree structure may imply that populations have been isolated from one another following a divergence (Sherry and Batzer 1997). Newer methods incorporate the effects of migration in tree displays (Pickrell and Pritchard 2012).

Perhaps the most commonly used method for displaying genetic variation among populations or individuals is principal component analysis (PCA). PCA is widely used in many areas of bioinformatics and is explained in more detail in Chapters 14 and 18. Briefly, PCA is a multivariate statistical technique in which an axis (a principal component [PC]) is projected through a matrix of distances in an effort to capture the maximum amount of variation in a single axis or line (reflected by the score of each population on the axis). This procedure is essentially a multivariate version of a regression analysis. After accounting for this first PC, a second axis, independent of the first one, is projected through the remaining variation in the matrix. The PCs can be plotted against one another (as shown in Figure 15.1) to display the variation captured in two dimensions. Often, the first two PCs reflect geographic locations of populations quite well, and formal methods have been designed to assess the degree of fit between genetic and geographic distances (Wang et al. 2012). Additional PCs can be examined to assess further aspects of genetic variation among populations or individuals. PCA yields results similar to those of multidimensional scaling, which is implemented in the popular PLINK software package (Purcell et al. 2007).

In its earliest applications, PCA was confined to population-level comparisons because typically only a few dozen loci could be genotyped. With so few loci, a high level of sampling variance existed, but this could be minimized by combining individuals together in pre-defined populations. Designating a priori population membership introduces bias, however, and it is preferable to analyze variation at the individual level. This is now achieved through the use of large-scale single nucleotide polymorphism (SNP) microarrays or whole genome sequencing,

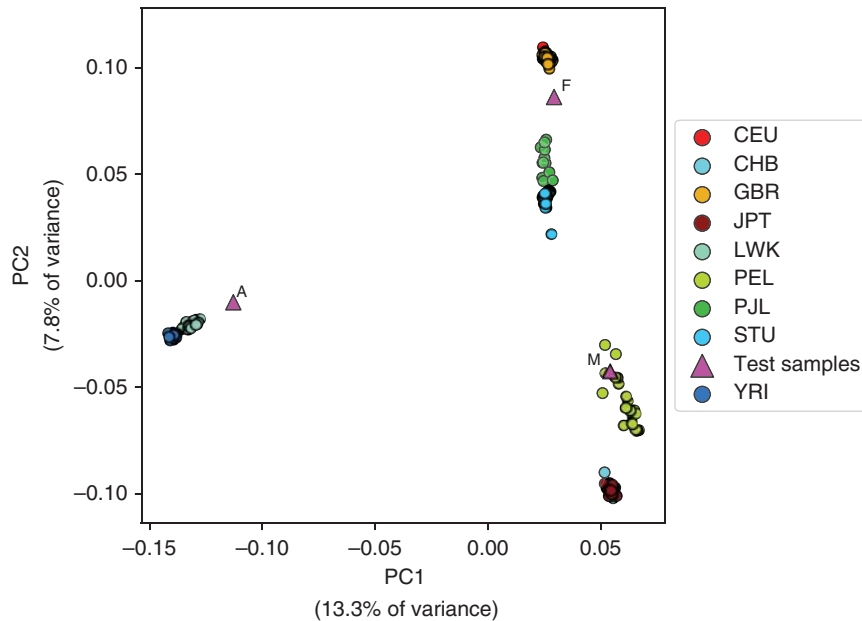


Figure 15.1 Principal components analysis (PCA) of nine world populations and three test samples. Individuals of similar ancestry show tight population clustering. The three test samples (large triangles) are placed in different locations on the plot, reflecting different ancestry for each of the samples. Sample A has genetic affinity with the African reference samples, sample F is similar to the European samples, and sample M is clustered among the South American samples. The reference populations are Africans: Kenyans (LWK), Nigerians (YRI); Europeans: CEPH with European ancestry (CEU), Great Britons (GBR); South Asians: Pakistanis (PJT), Sri Lankans (STU); East Asians: Han Chinese (CHB), Japanese (JPT); and South Americans: Peruvians (PEL).

in which thousands to millions of SNVs can be assessed in each individual (as in Figure 15.1). In general, individuals group according to their population affiliation, but there is often overlap between defined populations, especially among geographically contiguous or admixed populations (Rosenberg et al. 2005). PCA results can be influenced by factors such as linkage disequilibrium (Box 15.1) and ascertainment biases in microarray data (Albrechtsen et al. 2010a), so appropriate precautions must be taken in designing and interpreting PCA studies.

Because PCA can detect genetic similarities among individuals and populations, it is also widely used in genome-wide association studies (GWAS) as a tool to detect and adjust for population stratification in cases and controls. Using genome-wide SNV data, the popular EIGENSTRAT tool (Price et al. 2006) performs PCA on case and control individuals and enables users to eliminate or adjust for genetic outliers that could cause spurious gene–disease associations.

Demographic History Inference

The display methods just discussed provide a useful picture of genetic variation, but they are not necessarily informative (and can even be misleading) about migration events or changes in population size (Novembre and Stephens 2008). Furthermore, microarray-based SNVs, which have often been used in analyses of population genetic variation, are biased because they were typically selected for relatively high frequency ($> \sim 0.10$) in populations of interest for genome-wide association studies (usually Europeans) (Lachance and Tishkoff 2013a). Rare variants, which are highly informative for studies of population history, are vastly under-represented in microarray datasets, as are many variants found in non-European populations (Rosenberg et al. 2010). In the past several years, several major population surveys have been undertaken to obtain unbiased depictions of whole genome sequence variation: the 1000 Genomes Project (Auton et al. 2015; Sudmant et al. 2015a), the Simons Genome Diversity Project (Sudmant et al. 2015b; Mallick et al. 2016), the UK10K Project

(UK10K Consortium et al. 2015), and the Exome Aggregation Consortium (ExAc) and Genome Aggregation Database (gnomAD) (Lek et al. 2016). The UK10K Project and the ExAc and gnomAD databases focus primarily on individuals with disease phenotypes. With DNA sequence data in hand, methods can be used that take advantage of the full range of genetic variation, enabling a much richer portrait of human population history (Box 15.2).

Box 15.2 Inferring Demographic History

The multiple sequentially Markovian coalescent (MSMC) method can be used to infer effective population size using whole genome sequence data from one or more individuals. Starting with a mapped whole genome Binary Alignment Map (BAM) file, variants are first called using Samtools, BCFtools, and the bamCaller.py script. The output is a Variant Call Format (VCF) file and a mask file (indicating usable regions) for the sample. It is necessary to provide a reference sequence and the average sequencing depth of the sample. If the genotypes are to be phased (recommended), the data should be split into chromosomes. An example of a command to generate the single nucleotide polymorphism (SNP) calls and the sample mask file for chromosome 22 is as follows.

```
>samtools mpileup -q 20 -Q 20 -C 50 -u -r chr22 -f myReferenceGenome
myBamFile.bam | bcfutils call -c -v indels | bamCaller.py depthOfCoverage
myBamFile_chr22_mask.bed.gz | bgzip >chr22.vcf.gz
```

In this example, mpileup (part of Samtools) is run with the map quality (-q) and minimum per base alignment quality (-Q) set to 20, and the adjusted map quality (-C) set to 50. The uncompressed output (-u) is piped to BCFtools for calling. The consensus caller (-c) is used, variant sites (-v) are emitted, but insertion/deletions are excluded (-v indels). The output is piped to the bamCaller.py script, which generates a VCF file and an accompanying mask file that indicates the variant positions that are sufficiently covered to be used by MSMC.

After generating the genotype calls for each chromosome, the data can be phased using the SHAPEIT2 program. An individual sample may be phased using a reference panel that matches the input sample's population, or the sample may be phased together with 10 or more samples from the same population. The second option is recommended if the test population is not well matched to an existing reference population. Each chromosome must be processed separately. It is important to remove multi-allelic SNP sites from the VCF file prior to running SHAPEIT2. A genetic map for each chromosome should be used to adjust for differences in the recombination frequency along the chromosome. An example for phasing chromosome 22 in a group of 10 Tibetan samples previously called and combined into a single VCF file is shown.

```
>shapeit --input-vcf Ten_Tibetans_chr22_bamCaller.vcf
-M genetic_map_chr22.txt -O Ten_Tibetans_chr22.phased -T 1 --aligned

>shapeit -convert --input-haps Tibetan_chr22.phased --output-vcf
Tibetan_chr22.phased.vcf
```

The phased data are used as input into the generate_multihetsep.py script to create MSMC input files for each chromosome. Specific samples to be used for MSMC input can be extracted from the phased VCF files using BCFtools. It may be necessary to update (reheader) the VCF file produced by SHAPEIT2 before extracting the samples.

Once extracted, the sample VCF file and the corresponding sample mask file serve as the input files for the generate_multihetsep.py script. The generate_multihetsep.py script also takes as input a chromosomal mapping mask that specifies the uniquely mapping regions on that chromosome. A mapping mask file can be created by the user (see evomics.org/learning/population-and-speciation-genomics/2018-population-and-speciation-genomics/psmc-msmc-activity/ for additional information). An example

for generating an MSMC input file for two individuals (four haplotypes) from two different populations is shown below. Note that each sample requires the phased VCF data and the sample mask, but one mapping mask can be used for all samples. Prepare one file for each chromosome.

```
>generate_multihetsep.py \
--mask Tibetan1_chr22.bed.txt.gz \
--mask Chinese1_chr22.bed.txt.gz \
--mask Unique_mapping_mask_chr22.bed.txt.gz \
Tibetan1_chr22.vcf.gz Chinese1_chr22.vcf.gz \ >msmc_input_chr22.txt
```

The output of `generate_multihetsep.py` contains a list of segregating sites, the number of bases between the sites, and the phased alleles for the four haplotypes.

The MSMC program is now used to estimate the scaled effective population size and the cross-coalescence rate. When the input files contain samples from two populations, the cross-coalescence rate can be used to estimate the relative separation between the populations on a scale of 0–1, where 0 is complete separation and 1 is no separation. Add the `--skipAmbiguous` and the `-P` flag to calculate the relative cross-coalescence rate (e.g. `-P 0,0,1,1` where 0 and 1 identify the haplotype associated with each population). Use the files generated for each chromosome as input into the MSMC program.

```
>msmc --fixRecombination --skipAmbiguous -P 0,0,1,1 -t 12 -o my_msmc_output
msmc_input_chr1.txt msmc_input_chr2.txt ... msmc_input_chr22.txt
```

The final MSMC output file (see Figure 15.3) from this two-sample, four-haplotype run contains the estimates of the coalescent rate for the first population (`lambda_00`), the coalescent rate for the second population (`lambda_11`), and the coalescent rate across the two populations (`lambda_01`). The relative cross-population coalescent rate is calculated as $(2 * \text{lambda_01}) / (\text{lambda_00} + \text{lambda_11})$.

The output file also shows the time interval for each estimate. Time estimates and coalescent rates are scaled by the mutation rate. Dividing the scaled time by the mutation rate (e.g. $\mu = 1.25 \times 10^{-8}$ mutations/site/generation) yields the number of generations. Multiplying the number of generations by the generation time (e.g. 30 years/generation) produces a final time estimate in years. The inverse of the coalescent rate reflects the effective population size (N_e) and is scaled by the mutation rate. The actual effective population size can be calculated from the scaled coalescent rate as $N_e = (1/\text{lambda_00})/\mu$.

This example serves as a general guide. The MSMC program and accessory scripts have additional features and options that can be used to fine-tune performance and optimize accuracy. Additional improvements to MSMC demographic modeling are expected in MSMC version 2. Since MSMC output values are scaled, the values used for the mutation rate and the generation time will affect the final estimate of the effective population size. It is recommended that the confidence interval around these parameters be considered when evaluating the results.

For example, approaches have been devised to use these large-scale sequence collections to infer major demographic events such as migration, population bottlenecks, and population expansions. Many of these methods make use of the concept of coalescence (Rosenberg and Nordborg 2002). To understand the coalescence concept, imagine that DNA sequences have been obtained for a small chromosome region in two individuals. If these sequences differ at five nucleotide positions, we can infer that at least five mutations have occurred since the sequence was transmitted by the common ancestor of the two individuals. Because mutation is a regular, clock-like process within a species, we can use the mutation rate to estimate how long it would have taken for these five mutations to occur in the lineages that produced the two individuals. In this way, we can assign an approximate date to the common ancestor. This approach can be extended to the human population in general by comparing DNA

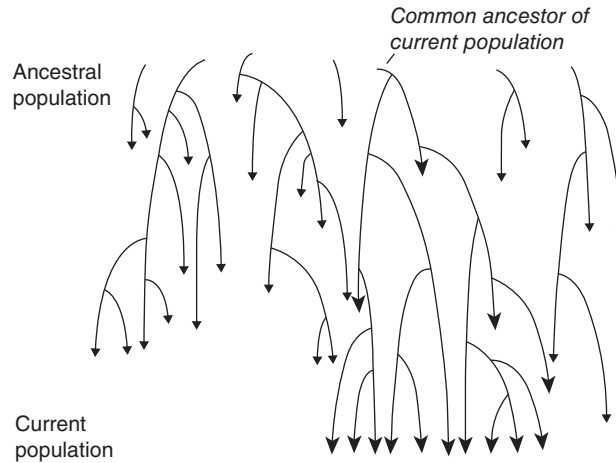


Figure 15.2 The coalescent process. Although the ancestral population contains multiple individuals, all lineages but one eventually become extinct for any gene. Thus, all of the variation in the current population can ultimately be traced to one common ancestor (the coalescent).

sequences from a sample of individuals and working backward to the individual from whom the existing variation would have descended. This ancestral individual is termed the coalescent. Figure 15.2, which illustrates the coalescent concept, shows that all ancestral lineages but one will eventually become extinct. In general, the larger the population, the longer this extinction process takes, and the further back in time the coalescent occurs. Thus, there is a direct relationship between population size and the date of the coalescent.

Methods have been developed to use the coalescent process to estimate the history of population sizes from genome sequence data. For example, the pairwise sequentially Markovian coalescent (PSMC) model compares the two DNA sequences in a single diploid individual using a hidden Markov model to move along the DNA sequences, measuring haplotype differences to make estimates of the coalescence time (Li and Durbin 2011). Because this method is limited to a single individual, most coalescence dates are older than 20 000 years. This approach was subsequently extended to multiple individuals (multiple sequentially Markovian coalescent [MSMC] method) (Schiffels and Durbin 2014), which allows inference of more recent population history. MSMC requires phased sequence data (see Glossary) and incorporates the effects of recombination. In addition, because individuals from different populations can be analyzed and compared, it is possible to estimate a cross-coalescence, which is a proxy for migration rate. An example of human population history estimated by MSMC is given in Figure 15.3. (See also Box 15.2 for details.)

Other methods for population history inference make use of the site (or allele) frequency spectrum described earlier. One of the most popular of these, diffusion approximations for demographic inference, or *dadi* (Gutenkunst et al. 2009), estimates parameters such as population sizes, divergence times, admixture events, and migration rates using partial differential equations to derive the allele frequency spectrum. Because of its multiple parameter estimates, *dadi* can be computationally demanding, and it is limited to three populations, each containing only a small number of individuals. Other methods have been developed to extend and improve this approach. (Schraiber and Akey 2015; Novembre and Peter 2016).

These methods, especially when applied to whole genome sequence data, have yielded many key insights about human evolutionary history (Novembre and Ramachandran 2011; Veeramah and Hammer 2014; Auton et al. 2015; Mallick et al. 2016; Nielsen et al. 2017). Broadly, they support a model in which anatomically modern humans arose first in Africa at least 200 000 years ago, where they accumulated a rich reservoir of genetic diversity. A subset of this population began to radiate out of Africa approximately 100 000 years ago, mostly replacing archaic humans, such as Neanderthals, in other parts of the world. As humans migrated throughout the world, they experienced successive reductions in population size (a serial founder effect), resulting in a strong negative correlation between genetic diversity

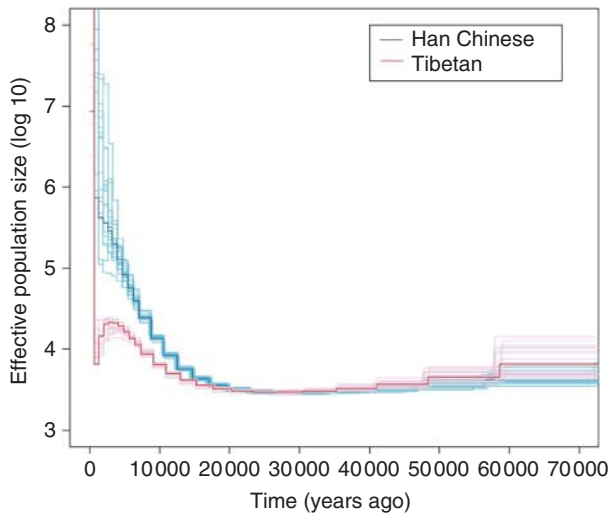


Figure 15.3 Multiple sequentially Markovian coalescent (MSMC) estimate of population histories. The effective population size as a function of time in the past for Tibetans (red) and Han Chinese (blue), estimated using four Han Chinese and four Tibetan genomes with greater than 99% corresponding genetic ancestries (Hu et al. 2017).

and distance from the African origin (Ramachandran et al. 2005). In addition to providing an overview of worldwide human demographic history, genetic studies have yielded detailed portraits of continental and individual human populations, such as those of Africa (Campbell et al. 2014; Beltrame et al. 2016), Asia (Abdulla et al. 2009; Liu et al. 2017), Australia (Malaspina et al. 2016), Europe (Fu et al. 2016; Gunther and Jakobsson 2016), the Indian subcontinent (Reich et al. 2009), Oceania (Duggan and Stoneking 2014), and the Americas (Skoglund and Reich 2016).

Although it is quite clear that anatomically modern humans largely replaced other hominid species as they migrated out of Africa, studies of ancient Neanderthal DNA sequences demonstrate that the genomes of all non-Africans studied thus far contain approximately 2% Neanderthal DNA. This reflects a small degree of ancient admixture (Sankararaman et al. 2014; Nielsen et al. 2017). The uniformity of the admixture level among non-Africans suggests that most of the mixture took place early in the radiation of humans out of Africa. Furthermore, there is evidence that some Neanderthal genetic variants, including those involving the immune response and skin pigmentation, provided selective advantages as modern human populations adapted to their new environments (adaptive introgression; Racimo et al. 2015; Dannemann and Kelso 2017). Much Neanderthal variation, however, appears to have undergone negative selection and was eliminated in modern human genomes (Sankararaman et al. 2014). Some modern humans also admixed with Denisovans, an ancient sister species of Neanderthals (Reich et al. 2010). Approximately 3–6% of each DNA sequence from modern Melanesians, Papuans, and Australians, and approximately 0.2% of each DNA sequence from East Asians, are of Denisovan origin (Racimo et al. 2015).

Admixture and Ancestry Estimation

Humans have a long and complex history of migration, gene flow, and population mixture (Hellenthal et al. 2014). Large-scale genomic data have made it possible to estimate the extent and timing of these events and their effects on individual ancestral composition. An early method, called STRUCTURE (Pritchard et al. 2000), uses a Bayesian Markov chain Monte Carlo (MCMC) algorithm to detect groups of individuals in whom Hardy–Weinberg equilibrium (HWE) (Box 15.1) is maintained. (Deviations from HWE indicate that a population sample may contain multiple subgroups.) The optimal number of groups within a sample can be estimated, and, for each individual in a group, the proportion of ancestry derived from each

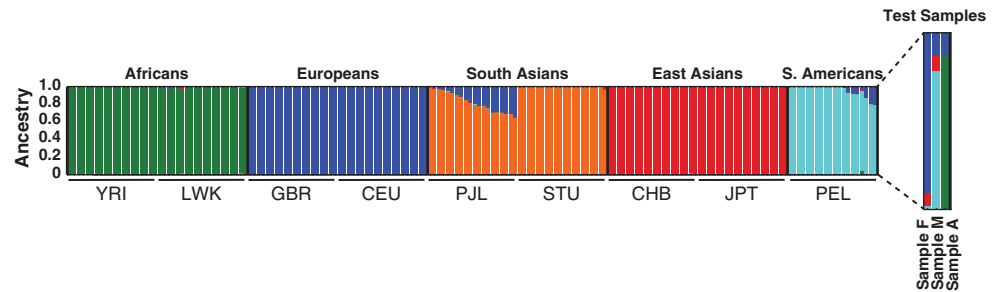


Figure 15.4 Admixture analysis of nine populations and three test samples. Individuals are displayed along the X-axis. The proportion of ancestry is shown on the Y-axis. Each individual is represented by a single bar. Five ancestry components are each indicated by a color. Each bar represents one individual and shows the fraction of ancestry for each of the ancestry clusters. Each of the three test samples has a predominant ancestry, yet all have some admixture as indicated by the different colors. Compare the ancestry estimates for the three test samples with the location of each sample in principal component analysis (Figure 15.1). For definitions of abbreviations, see the legend to Figure 15.1.

group is estimated. The timing of admixture events is not estimated, but the STRUCTURE program provides a useful display of ancestral composition. This method can become computationally demanding if the number of samples is large. Subsequent improvements, such as Frappe (Tang et al. 2005), ADMIXTURE (Alexander et al. 2009), and fastSTRUCTURE (Raj et al. 2014), provide increased computational speed and accuracy to detect population structure and estimate individual ancestry (Figure 15.4; and see below for details) (Liu et al. 2013). The Chromopainter and fineSTRUCTURE (Lawson et al. 2012) algorithms incorporate haplotype data to improve the accuracy of detection of fine population structure ($F_{ST} < 0.01$) and ancestry estimation, at the expense of some increase in computational time. The program known as fineSTRUCTURE has been used, for example, to detect population structure even within relatively homogeneous populations such as those of Great Britain (Leslie et al. 2015), where the average F_{ST} value is less than 0.001.

Other tests of admixture take advantage of the fact that gene flow produces individuals in whom large chromosome segments are derived from more than one ancestral population. Because of recombination, these chromosomal segments or “haplotype blocks” will become shorter with time; therefore, the average length of these blocks provides a means of dating major admixture events. This principle has been incorporated in the programs ROLLOFF (Moorjani et al. 2011) and GLOBETROTTER (Hellenthal et al. 2014), which have been used to date major migration events such as the Bantu expansion in Africa, the Mongol expansion in Eurasia, and relatively recent north African gene flow into southern Europe (Hellenthal et al. 2014).

Following is a more detailed example to illustrate how admixture analysis and PCA can be used in ancestry analysis of three specific individuals wanting to know more about their ethnic ancestry. This is the kind of analysis commonly performed by many commercial DNA testing companies. A typical starting point for this type of ethnic ancestry analysis is a multi-ethnic Variant Call Format (VCF) file containing genotypes in rows and individuals in columns. Whole genome sequencing (WGS) data from 26 world populations are readily available from the 1000 Genomes Project. Using this resource, reference populations can be selected for the analysis. New study samples can then be merged into the reference VCF file to examine the genetic relationships among all individuals and to assess potential stratification issues for case-control studies.

The tools needed for this example are VCFtools or BCFtools, PLINK and PLINK2, EIGENSOFT (version EIG-6.1.4), ADMIXTURE, and a plotting package of your choice (Excel, R, etc.). These programs typically need to be installed on a modern, multi-core LINUX or UNIX system. There are many high-quality software packages available for PCA. This example uses these particular tools because they are well established, optimized for genetic data, and have additional functionality for file manipulation, case-control studies, and hypothesis testing.

This ancestry analysis example uses a VCF file created from whole genome sequencing of nine populations from five distinct regions of the world. The sample populations are *Africans* (Nigerians [YRI] and Kenyans [LWK]), *Europeans* (Utah CEPH [CEU] and Great Britons [GBR]), *East Asians* (Chinese [CHB] and Japanese [JPT]), *South Asians* (Pakistanis [PJL] and Sri Lankans [STU]), and *South Americans* (Peruvians [PEL]). There are 20 samples per population. This dataset was assembled from the 1000 Genomes WGS data and filtered to contain approximately ~6.77 million common ($\text{maf} \geq 0.05$) SNPs. A second VCF file contains three test samples of unknown ancestry. The two VCF files can be downloaded from www.wiley.com/go/baxevanis/Bioinformatics_4e.

The two VCF files are first indexed and merged using BCFtools to create a single compressed VCF file. Once this is done, the VCF file must be converted to PLINK binary format. PLINK provides a convenient and rapid format for filtering samples and markers. For most steps, PLINK2 is used for better performance.

```
>bcftools index world_samples.vcf.gz
>bcftools index test_samples.vcf.gz
>bcftools merge -Oz world_samples.vcf.gz test_samples.vcf.gz -o merged.vcf.gz
>plink2 -vcf merged.vcf.gz --input-missing-phenotype -9 --make-bed --out merged
```

Next, the data must be “cleaned” so that all samples and variants have minimal missing data and well-supported genotype calls. These settings can vary but, in general, the best results are obtained when all samples and all loci have less than 10% missing data. For most applications, the variants that deviate *strongly* from HWE should be removed. PLINK2 can be used to perform these three filtering steps in a single command, given below. Only the merged binary PLINK file (bfile) is required in the command.

```
>plink2 --bfile merged --mind 0.05 --geno 0.05 --hwe 0.001 --make-bed --out
merged_cleaned
```

Many population models assume that genetic markers are independently segregating. Therefore, it is important to remove closely linked markers that are in strong linkage disequilibrium. This step also reduces redundant genetic information and the size of the dataset. Using PLINK2 and the commands given below, markers in a 50 kb sliding window are first identified if their pairwise correlation (r^2) exceeds 0.1. The second step extracts the non-correlated markers.

```
>plink2 --bfile merged_cleaned --indep-pairwise 50kb 1 0.1
>plink2 --bfile merged_cleaned --extract plink2.prune.in --make-bed -out
merged_cleaned_pruned
```

For the dataset used here, one will notice a sizeable reduction in the number of SNPs: ~6.77 million starting loci will be filtered to ~140 000 unlinked loci. In general, a minimum of ~100 000 SNPs is recommended for full genome analysis and case-control p value adjustments; however, many population structure questions may be addressed with as few as 10–20 000 unlinked variants if the populations are relatively distinct. The data are now ready for PCA and admixture analysis.

For this example, the EIGENSOFT package will be used to perform a PCA. The cleaned and pruned data can be exported to a standard linkage file using PLINK. With the command given below, a pedigree (linkage) file (.ped) and a map file (.map) can be created. The missing phenotype value (–9) added earlier can be changed in place to the unaffected phenotype value (1) using the UNIX sed command.

```
>plink --bfile merged_cleaned_pruned --recode --out pca_data
>sed -i `s/-9/1/' pca_data.ped
```

The pedigree file can now be converted to EIGENSTRAT format. Copy the parameter file from the convertf folder of the EIGENSOFT package to the current directory and edit the file as shown in the example to convert a pedigree (.ped) file to an EIGENSTRAT file. The `pca_data.ped` and `pca_data.map` files can be generated by passing the parameter file to CONVERTF using the following command.

```
>convertf -p par.PED.EIGENSTRAT
```

The `smart_pca.perl` script is now used to call SMARTPCA and perform the PCA analysis. The number of shared alleles at every locus will be assessed for all pairwise combinations of individuals. This will create an allele-sharing covariance matrix for all possible pairwise combinations of samples, which can be used to create the output files. The three output files from the CONVERTF program can be used as inputs into the `smartpca.perl` script. The options for `smartpca.perl` specify the converted genotype data (`-i`), the converted SNP data (`-a`), the converted sample information (`-b`), the number of PCs to output (`-k`), an output file of PCs (`-o`), a plot of PC1 and PC2 (`-p`), the eigenvalues for all individuals and PCs (`-e`), the logfile (`-l`), and an outlier switch (`-m`) set to 0 to prevent outlier removal. If one is doing case-control studies, the program can be used to identify samples that are statistically different from other samples in the dataset.

```
>smartpca.perl -i pca_data.eigenstratgeno -a pca_data.snp -b pca_data.ind
-k 12 -o pca_data.pca -p pca_data.plot -e pca_data.eval -l pca_data.log -m 0
```

The eigenvectors can be examined and plotted using the data found in the `pca_data.pca.evec` (or `pca_data.pca`) file. The data for each sample and the percentage of the variance accounted for by each of the PCs 1–12 are shown in the file. Note that the first PC captures the highest percentage of the variance, and each subsequent PC captures less. The decrease typically follows a negative exponential curve, and the first several PCs are most informative for ancestry and population structure. By plotting PC1 and PC2, the relationships among the 180 world samples and the three test samples can be visualized (Figure 15.1). Plotting other dimensions may also give additional insight into the population relationships.

The three test samples are located in different regions of the plot, indicating different ancestry for each of the samples. One sample is located near the YRI and LWK groups, indicating mostly African ancestry, the second sample has predominantly northern European ancestry, and the third is grouped with South Americans, likely indicating Native American ancestry.

The following steps demonstrate how ADMIXTURE can be used to estimate the proportion of ancestry of each individual in our current dataset. Return to the binary PLINK file that has been cleaned, filtered, and pruned. This file will be the input for ADMIXTURE. Because the individuals in the data are from five distinct well-separated geographic regions, a value of $K = 5$ is a reasonable number of population clusters for this analysis. It is recommended that a most likely value of K be estimated directly from the data, that all values $K = 2 \dots 10$ be examined, and that all additional population information be considered when choosing the optimal number of clusters for a given dataset. Using a large number of markers (>100 000) improves the accuracy of the ancestry estimates, especially when the populations are closely related. In this dataset there are sufficient markers to achieve good ancestry estimates. The ADMIXTURE program can be run using the final binary pedigree file, `merged_cleaned_pruned`, as the input.

```
>admixture merged_cleaned_pruned.bed 5
```

ADMIXTURE produces two output files. The `merged_cleaned_pruned.5.Q` file contains the point estimates of ancestry for each sample. Confidence intervals for the point estimates can be obtained by adding `-B` to the command. For clarity, sample names can be added to the ancestry point estimates file using the names from the pedigree file used for PCA using the commands below.

```
>cut -f2 -d' ' pca_data.ped >names
>paste -d' ' names merged_cleaned_pruned.5.Q > ancestry.5.Q
```

Population samples in this file are ordered alphabetically by the population identifiers (e.g. CEU for Europe, CHB for East Asia, GBR for Europe, and so forth). Each row represents one sample. There are five ancestry columns corresponding to the fraction of ancestry estimated for each of the five clusters. The row values sum to 1.00. The first 20 samples have ancestry estimates that exceed 0.99 in one of the columns; these are all CEU samples. The GBR samples (rows 41–60) also have high estimated ancestry in this same column (cluster). Thus, this column represents the cluster most closely associated with European ancestry. Examining the

East Asian CHB and JPT samples will allow one to locate the column (cluster) that is associated with East Asian ancestry. This process can be repeated for each of the populations.

The table of ancestry results can be more conveniently visualized by plotting all samples as stacked bars in a single plot (Figure 15.4). At the end of the plot are the three test samples, enlarged for clarity. Sample A has ~87% African ancestry and ~13% European ancestry. This individual self-identifies as African–American, and the ancestry result is not unexpected. Many African–Americans have some European–American admixture. Sample F is a Finnish individual and is most similar to the other Europeans but also has about ~9% ancestry that does not cluster with the European samples used in this dataset. Sample M self-identifies as Mexican, and shares ~79% ancestry with the South American reference samples but also has ~13% ancestry that is attributable to European admixture and ~8% Asian ancestry. Finally, compare the results of the model-based approach with the PCA. Note how genetic admixture can place samples in intermediate positions between two non-admixed reference populations in the PCA. For example, sample A, an African–American individual, is positioned between the reference African and European clusters.

Detection of Natural Selection

Many or most regions of the genome are thought to have little or no functional significance and are therefore selectively neutral. The major forces affecting their variation are mutations (which introduce novel variants) and genetic drift (which may eliminate them or increase their frequency owing to stochastic variation in small populations). In contrast, coding and regulatory regions of the genome are potentially subject to natural selection because of their roles in maintaining important functions. Natural selection exists in several basic forms. Positive selection, which is considered the major force in adaptive evolution, increases the frequencies of variants that confer a survival or reproductive advantage. Negative selection, also termed purifying selection, is exerted against deleterious variants. Many harmful variants are removed by natural selection before they can achieve an appreciable allele frequency. This form of negative selection, termed background selection, produces regions of the genome that tend to lack variation and can be highly conserved across species (Vitti et al. 2013).

Positive selection can result in a selective sweep, in which a new, adaptive variant arises and rapidly increases to high frequency or fixation because of its selective advantage. This scenario is often termed a hard sweep and is relatively easy to detect in genomic data using the methods discussed below. Hard sweeps, however, appear to be relatively rare in human populations (Pritchard et al. 2010). In contrast to the classic hard sweep scenario, new variants may be selectively neutral at first and can increase in frequency because of genetic drift. If the variant (or variants) later becomes selectively advantageous, this is termed selection on standing variation. Owing to their longer history and the potential for recombination, standing variants can sometimes be found on different haplotype backgrounds. Selection on this form of standing variation is termed a soft sweep (Pritchard et al. 2010). Because of its relative complexity, a soft sweep is more difficult to detect in genomic data than is a hard sweep (Teshima et al. 2006).

Whereas positive selection produces alleles of high frequency and negative selection yields alleles of low frequency, another form of selection, termed balancing selection, tends to maintain alleles of intermediate frequency. This can occur when heterozygotes enjoy a selective advantage compared with both forms of homozygotes. A classic example of heterozygote advantage (also known as over-dominance) is given by sickle cell disease in malarial environments. This recessive condition is caused by a specific amino acid change in the β -globin (also known as hemoglobin beta) locus and is usually fatal if not treated (Rees et al. 2010). Thus, homozygotes are strongly selected against. However, heterozygotes are 50–90% less likely to contract severe *Plasmodium falciparum* malaria than are normal homozygotes because their erythrocytes are inhospitable to the malarial parasite (Bunn 2013). Thus, the sickle cell-causing variant increases in frequency, but only to a certain degree, because at

too high a frequency the burden of sickle cell disease in the population would outweigh the anti-malarial advantage. Accordingly, the allele frequency of the sickle cell variant can be as high as 0.15–0.20 in malarial environments (Piel et al. 2010).

Balancing selection favoring allelic diversity is also thought to be largely responsible for the high levels of genetic variation seen in genes involved in the immune response, such as the major histocompatibility complex and the ABO blood group (Hughes and Yeager 1998; Key et al. 2014). Balancing selection on some immune-response genes has persisted for millions of years, resulting in polymorphisms that are shared among humans and other apes (trans-species polymorphisms) (Azevedo et al. 2015).

Many methods have been devised to detect different types of natural selection (Fu and Akey 2013; Vitti et al. 2013; Fan et al. 2016). Relatively recent selection in human populations can be detected by relatively straightforward comparisons of allele frequency differences among populations. For example, an elevated locus-specific F_{ST} value, relative to the genome average across all loci, can be an indication of natural selection, as in the case of skin pigmentation genes discussed earlier. The population branch statistic (PBS) (Yi et al. 2010) is a related test that compares the branch lengths of population trees on a gene-by-gene basis to search for genes in which one population has an unusually long branch length, indicating substantial divergence from other populations.

Another class of selection statistics is based on the fact that rapid positive selection increases the frequency of the selected variant as well as nearby linked variants. Thus, there is an increased level of linkage disequilibrium in the region. In effect, haplotypes consisting of multiple linked alleles are increased in frequency because of selection. Therefore, many individuals in the population will have a relatively long region of complete (or nearly complete) homozygosity because they will have two identical copies of the selected haplotype. Under neutrality, such haplotypes would soon break down due to recombination, but positive selection increases their frequency rapidly enough to outpace the effects of recombination. Methods such as the extended haplotype homozygosity (EHH) test (Sabeti et al. 2002) and integrated haplotype score (iHS) (Voight et al. 2006) search for regions in which tracts of homozygosity are larger than would be expected in the absence of selection. A variation on these tests, termed the cross-population extended haplotype homozygosity (XP-EHH) statistic (Sabeti et al. 2007), compares haplotype lengths between pairs of populations. The EHH and iHS tests are especially capable of detecting incomplete selective sweeps (i.e. the selected allele has not yet reached a frequency of 1.0), while XP-EHH can detect a haplotype that has reached fixation in one population but not in another. These methods are commonly employed on genome-wide datasets (either microarray SNVs or whole genome sequences), so a challenge is to determine whether a putative selected region is functionally significant or merely represents the upper tail of a statistical distribution of haplotype lengths. Another challenge is that the selected region may contain multiple genes, any of which could contain the actual variant under selection. In general, these methods are more suitable for detecting hard sweeps than soft sweeps (Vitti et al. 2013).

Several methods have been designed to take advantage of sequence data to detect more subtle selective events like soft sweeps and selection on polygenic traits (e.g. height). For example, the singleton density score (SDS) (Field et al. 2016) exploits the fact that chromosome regions near selected variants have lower frequencies of linked singleton alleles (a singleton allele occurs only once in the population). Compared with methods such as iHS, SDS has substantial statistical power to detect very recent selection on standing variation or on polygenes. When applied to a large British sequence dataset (UK10K Project), SDS demonstrated selection for light hair, blue eye color, and increased height in the past 2000–3000 years (Field et al. 2016). Identity-by-descent methods, which identify selected DNA segments inherited from a common ancestor, also have increased power to detect selection on standing variation (Albrechtsen et al. 2010b).

Finally, some selection–detection approaches have amalgamated several existing tests to increase the power and accuracy of signal detection. One of the most popular, the Composite

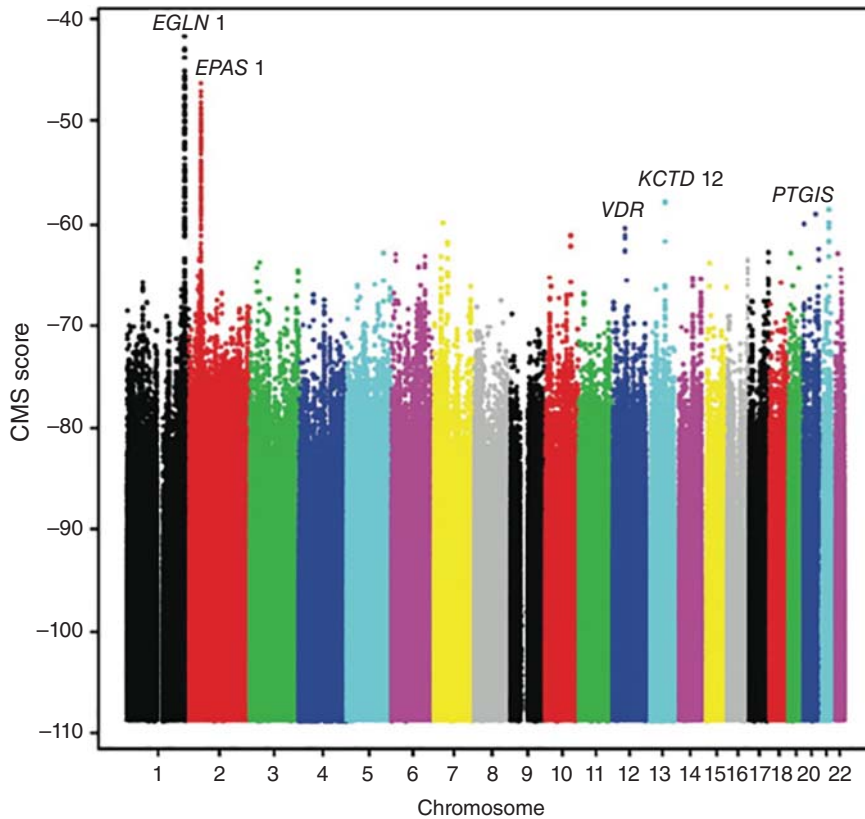


Figure 15.5 A Manhattan plot of Composite of Multiple Signals (CMS) scores (Y-axis) across the 22 autosomes (X-axis) (Hu et al. 2017). Each dot represents one single nucleotide variant. This plot, based on whole genome sequence data collected from Tibetans, shows high levels of positive selection on the *EGLN1* and *EPAS1* genes (see text), in addition to several other genes, including the vitamin D receptor gene (*VDR*).

of Multiple Signals (CMS) test (Grossman et al. 2010), combines the iHS , XP-EHH, and F_{ST} tests with two additional tests (one based on frequencies of newly arisen, “derived” alleles and one based on absolute length of haplotypes). An advantage of CMS is that it can localize the selected variant much more precisely than any of the single tests. When applied to the 1000 Genomes Project dataset, CMS identified a number of new regions likely to be under selection, including a number of innate immune-response genes (Grossman et al. 2013). Because these tests are genome wide in their coverage, it is common to graph the results as a “Manhattan plot,” similar to the graphs generated in a genome-wide association study (Figure 15.5). As in studies of disease-causing variants, it is critical to demonstrate the functional significance of putative selected variants through *in vitro* experiments and studies of animal models (Lachance and Tishkoff 2013b).

These methods have been used to detect the effects of natural selection on a number of human genes, many of which are related to disease resistance (see Table 15.1 for examples). Because different environments harbor different pathogens, it is not surprising that many selected genes encode components of the immune response (Quintana-Murci 2016). Malaria, which still causes up to 1 million deaths per year (Reiff and Striepen 2009), provides a classic example of selection for disease resistance and has been a selective agent in tropical environments for 10 000–20 000 years (Volkman et al. 2001). As discussed previously, *P. falciparum* has resulted in balancing selection on several genes that can also cause disease (Table 15.1). *Plasmodium vivax*, another cause of malaria, has exerted strong selection on the Duffy blood group locus, which encodes a chemokine receptor on erythrocyte surfaces that is used by *P. vivax* to enter the cells. A null allele at this locus, and the consequent absence of the receptor,

Table 15.1 Examples of genes that have undergone natural selection in human populations.

Adaptation	Gene (gene product)	Reference
Malaria resistance	<i>HBB</i> (β -globin)	Allison 1954; Kwiatkowski 2005
	<i>HBA1</i> (α -globin)	Flint et al. 1986
	<i>G6PD</i> (glucose 6 phosphate dehydrogenase)	Tishkoff et al. 2001; Sabeti et al. 2002
	<i>FY</i> (Duffy blood group)	Hamblin and Di Rienzo 2000; Hamblin et al. 2002
Melanin expression in skin in response to sunlight	<i>SLC24A5</i> (solute carrier; cation exchanger)	Lamason et al. 2005
	<i>SLC45A2/MAPT</i> (solute carrier)	Norton et al. 2007
	<i>OCA2</i> (melanosome membrane protein)	Donnelly et al. 2012
	<i>MC1R</i> (melanocortin 1 receptor)	Savage et al. 2008
Hereditary lactase persistence; ability to digest cow's milk in adulthood	<i>LCT</i> regulatory region (lactase expression)	Tishkoff et al. 2007
Ability to digest starch	<i>AMY1</i> (amylase gene copy number)	Perry et al. 2007
High-altitude hypoxia adaptation in Tibetans	<i>EPAS1</i> (<i>HIF2A</i> ; component of hypoxia-inducible factor [HIF] pathway)	Beall et al. 2010; Simonson et al. 2010; Yi et al. 2010
	<i>EGLN1</i> (PHD2; regulator of HIF pathway)	Simonson et al. 2010; Lorenzo et al. 2014
Trypanosome resistance in Africa	<i>APOL1</i> (apolipoprotein L1)	Genovese et al. 2010
Adaptation to diet rich in omega-3 fatty acids in Inuits	<i>FADS2</i> (fatty acid desaturase)	Fumagalli et al. 2015

Many other examples of natural selection in humans have been discovered or suggested, and this table lists only salient examples of genes under selection for each trait. For more complete lists, see Sturm and Duffy (2012), Scheinfeldt and Tishkoff (2013), Vitti et al. (2013), Fan et al. (2016), and Haas and Payseur (2016).

is highly protective against *P. vivax* and has a frequency of nearly 100% in most sub-Saharan African populations (Hamblin and Di Rienzo 2000). It is absent in other human populations. Additional disease-related examples of selection are listed in Table 15.1.

Changes in diet can also have strong selective effects. For example, infants worldwide produce lactase, an enzyme that allows them to metabolize lactose in their mothers' milk. In most populations, lactase expression is downregulated in adults, but cattle-herding populations in Europe and parts of Africa maintain lactase production through adulthood (hereditary lactase persistence). Selection on regulatory elements near the lactase gene (*LCT*) has occurred independently in European and African populations, providing an excellent example of convergent evolution in humans (Tishkoff et al. 2007). It has been estimated that the African variant arose approximately 5000 years ago, while the European variant arose 9000 years ago (Tishkoff et al. 2007; Fan et al. 2016). However, recent studies based on 230 ancient DNA samples suggest that the European variant arose more recently, about 4000 years ago (Mathieson et al. 2015).

Another good example of a dietary adaptation is an increase in the number of copies of the salivary amylase-encoding *AMY1* gene after the adoption of agriculture (Perry et al. 2007). Amylase facilitates the salivary hydrolysis of starch components in the types of food more common in agricultural populations and may help to protect against intestinal disease. DNA sequencing of wolf and dog genomes has demonstrated a similar increase in amylase copy number as dogs became domesticated and became dependent on human-generated agricultural products (Axelsson et al. 2013).

As humans have migrated throughout the world, they have adapted selectively to a large variety of different climates, latitudes, and altitudes (Jeong and Di Rienzo 2014). Variation in skin pigmentation is associated with latitude and is thought to reflect adaptation to variation in sunlight exposure (Sturm and Duffy 2012). As shown in Table 15.1, a number of genetic variants associated with pigmentation have been the targets of natural selection. It has long been thought that increased melanin pigment production in tropical environments protects against harmful ultraviolet radiation, while decreased melanin at higher latitudes may facilitate vitamin D production in low-light environments (Jablonski and Chaplin 2010). However, the vitamin D hypothesis remains controversial, and other mechanisms may help to account for variation in human skin pigmentation (Sturm and Duffy 2012).

Just as climatic factors such as sunlight vary with latitude, oxygen concentration varies with altitude. Because of oxygen's vital role in survival, it is not surprising that decreased oxygen availability at high altitude has resulted in strong natural selection in some high-altitude populations. Among the best-studied high-altitude populations are Tibetans, who have lived for thousands of years at altitudes of 4000 to almost 5000 m. Oxygen availability at this altitude is 40% less than at sea level. Non-adapted individuals frequently exhibit hypoxic responses such as high-altitude pulmonary or cerebral edema, pulmonary hypertension, and mountain sickness (Macinnis et al. 2010). Native Tibetans, who thrive at these altitudes, exhibit a suite of unique, heritable traits, including protection from polycythemia (proportionally increased erythrocyte number), elevated birth weights compared with non-high-altitude individuals, and increased arterial oxygen saturation (Beall 2007). Tibetan hemoglobin levels at high altitude are similar to those of non-adapted populations at sea level, allowing Tibetans to avoid harmful consequences of polycythemia. Genome-wide scans for natural selection, using approaches such as the iHS, XP-EHH, PBS, and CMS tests, have demonstrated that several genes, including *EPAS1* and *EGLN1*, have undergone strong positive selection in the Tibetan population (Beall et al. 2010; Simonson et al. 2010; Yi et al. 2010; Hu et al. 2017).

The *EPAS1* and *EGLN1* genes encode components of the hypoxia-inducible factor (HIF) pathway and have emerged as targets of natural selection in multiple genetic studies (reviewed in Simonson et al. 2015). Both genes contain variants that have high frequencies in Tibetans (>80%) but are virtually absent in neighboring populations, such as the Han Chinese. Given the critical role of the HIF pathway in erythropoiesis, these genes represent plausible candidates for the Tibetan high-altitude phenotype. The *EPAS1* haplotype that has undergone positive selection in Tibetans was likely contributed to Tibetan ancestors by Denisovans (Huerta-Sanchez et al. 2014), which is another example of adaptive introgression. A recent whole genome sequence analysis indicates that this is the only Denisovan gene that has undergone positive selection in Tibetans (Hu et al. 2017). *EGLN1* has undergone positive selection in both the Tibetan and high-altitude Andean populations (Bigham 2016), but, as with *LCT*, different variants have been selected in the two populations. Functional analysis of *EGLN1* expression in a cell culture system has demonstrated that the Tibetan-specific variants recapitulate the Tibetan phenotype of reduced erythropoiesis under hypoxic conditions (Lorenzo et al. 2014), a good example of functional validation of a statistically identified selection target.

Other Applications

In addition to illuminating our understanding of human evolutionary history, studies of genetic variation have several other important applications in forensic science, human genetics, and medicine. For example, population genetic studies have shown that traditional concepts of human “races” are simplistic and potentially misleading because of the complex history of migration and admixture in human populations (Jorde and Wooding 2004; Royal et al. 2010). Estimates of genetic ancestry at the individual level, such as those shown in Figure 15.4, help to avoid the misperception that humans can be accurately categorized into discrete, mutually exclusive categories.

Population genetic theory and methods have also been highly useful in the forensics arena, where genetic variation in populations must be assessed to derive accurate random match probabilities (i.e. the probability that someone else in a population could have the same genotype profile as the suspect under criminal investigation) (Kayser and De Knijff 2011). Population stratification, which can be analyzed using population genetic methods, plays an important role in accurately estimating these probabilities. In addition, the patterns of variation revealed in large-scale population genetic studies help to assess the appropriateness of reference populations used in forensic analysis.

Finally, population genetics has contributed substantially to the goal of identifying and characterizing disease-causing genes. Disease-causing variation is a subset of genetic variation in general, and the same evolutionary processes can affect both neutral and disease-causing variants. Population genetic concepts such as HWE and linkage disequilibrium are used routinely in searches for disease-causing genes (Manolio 2013; Visscher et al. 2017). Evolutionary conservation across species is used to assess the functional significance of both non-coding and coding DNA (Encode Project Consortium 2012) and to help assign pathogenicity scores to candidate disease-causing variants (Cooper and Shendure 2011). Studies of population genetic variation have helped to elucidate the roles of rare and common alleles in disease causation (Tennesen et al. 2012; Quintana-Murci 2016). Population genetic datasets such as the 1000 Genomes Project (Auton et al. 2015) provide invaluable information on the rarity and distribution of disease-causing candidates. They also help to estimate the extent to which genetic findings in one population can be applied to others (Rosenberg et al. 2010; Marigorta and Navarro 2013). It is fair to state that, without a good understanding of population genetics and variation, our ability to detect and understand disease-causing variation would be severely compromised.

Summary

Human population genetics has evolved considerably in the past several decades. Most theoretical and methodological advances have been driven by two factors: vast improvements in computational power and a wealth of detailed genetic data, now typified by whole genome sequences. Because of these changes, it is now possible to infer detailed aspects of population history, including population bottlenecks and expansions, migration events, and natural selection on both Mendelian and polygenic traits. Because of the complexity of human demographic history, it remains challenging to parse some of the more subtle aspects of human genetic evolution, such as soft selective sweeps and repeated admixture events. Nevertheless, substantial progress is being made in solving these challenges. With the steady accumulation of data, methods, and findings, population genetics will doubtless play an ever-increasing role in our understanding of human evolution, health, and disease.

Internet Resources

Ancestry and admixture estimation and dating

1000 Genomes Project	www.internationalgenome.org
ADMIXTOOLS	reich.hms.harvard.edu/software
ADMIXTURE	software.genetics.ucla.edu/admixture
BCFtools	samtools.github.io/bcftools
Frappe	med.stanford.edu/tanglab/software/frappe.html
GLOBETROTTER	paintmychromosomes.com
HAPMIX	www.stats.ox.ac.uk/~myers/software.html
TreeMix	web.stanford.edu/group/pritchardlab/software.html
VCFtools	vcftools.github.io/index.html

Detection of natural selection

CMS and CMS _{GW}	www.broadinstitute.org/cms/cms-composite-multiple-signals
MEGA7 (Tajima's D, HKA, etc.)	www.megasoftware.net
Singleton density score (SDS)	web.stanford.edu/group/pritchardlab/software.html
Selscan (integrated haplotype score [iHS] and cross-population extended haplotype homozygosity [XP-EHH])	hernandezlab.ucsf.edu/software

Population history inference

Dadi	bitbucket.org/gutenkunstlab/dadi
Multiple sequentially Markovian coalescent (MSMC)	github.com/stschiff/msmc github.com/stschiff/msmc2/releases
MSMC-tools	github.com/stschiff/msmc-tools
Pairwise sequentially Markovian coalescent (PSMC)	github.com/lh3/psmc
Samtools	www.htslib.org
SHAPEIT	mathgen.stats.ox.ac.uk/genetics_software/shapeit/shapeit.html

Population structure analysis

EIGENSOFT	www.hsph.harvard.edu/alkes-price/software
EIGENSTRAT	www.hsph.harvard.edu/alkes-price/software
fineSTRUCTURE	paintmychromosomes.com
PLINK	zzz.bwh.harvard.edu/plink
PLINK2	zzz.bwh.harvard.edu/plink/plink2.shtml
STRUCTURE, fastSTRUCTURE	web.stanford.edu/group/pritchardlab/structure.html

References

- Abdulla, M.A., Ahmed, I., Assawamakin, A. et al. (2009). Mapping human genetic diversity in Asia. *Science* 326: 1541–1545.
- Albrechtsen, A., Nielsen, F.C., and Nielsen, R. (2010a). Ascertainment biases in SNP chips affect measures of population divergence. *Mol. Biol. Evol.* 27 (11): 2534–2547.
- Albrechtsen, A., Moltke, I., and Nielsen, R. (2010b). Natural selection and the distribution of identity-by-descent in the human genome. *Genetics* 186: 295–308.
- Alexander, D.H., Novembre, J., and Lange, K. (2009). Fast model-based estimation of ancestry in unrelated individuals. *Genome Res.* 19: 1655–1664.
- Allison, A.C. (1954). Protection afforded by sickle-cell trait against subtertian malarial infection. *Br. Med. J.* 1: 290–294.
- Auton, A., Brooks, L.D., Durbin, R.M. et al. (2015). A global reference for human genetic variation. *Nature* 526: 68–74.
- Axelsson, E., Ratnakumar, A., Arendt, M.-L. et al. (2013). The genomic signature of dog domestication reveals adaptation to a starch-rich diet. *Nature* 495: 360–364.
- Azevedo, L., Serrano, C., Amorim, A., and Cooper, D.N. (2015). Trans-species polymorphism in humans and the great apes is generally maintained by balancing selection that modulates the host immune response. *Hum. Genomics* 9: 21.
- Beall, C.M. (2007). Two routes to functional adaptation: Tibetan and Andean high-altitude natives. *Proc. Natl. Acad. Sci. U.S.A.* 104 (Suppl 1): 8655–8660.
- Beall, C.M., Cavalleri, G.L., Deng, L. et al. (2010). Natural selection on EPAS1 (HIF2alpha) associated with low hemoglobin concentration in Tibetan highlanders. *Proc. Natl. Acad. Sci. U.S.A.* 107: 11459–11464.

- Beltrame, M.H., Rubel, M.A., and Tishkoff, S.A. (2016). Inferences of African evolutionary history from genomic data. *Curr. Opin. Genet. Dev.* 41: 159–166.
- Bigham, A.W. (2016). Genetics of human origin and evolution: high-altitude adaptations. *Curr. Opin. Genet. Dev.* 41: 8–13.
- Bunn, H.F. (2013). The triumph of good over evil: protection by the sickle gene against malaria. *Blood* 121: 20–25.
- Campbell, C.D. and Eichler, E.E. (2013). Properties and rates of germline mutations in humans. *Trends Genet.* 29 (10): 575–584.
- Campbell, M.C., Hirbo, J.B., Townsend, J.P., and Tishkoff, S.A. (2014). The peopling of the African continent and the diaspora into the new world. *Curr. Opin. Genet. Dev.* 29: 120–132.
- Conrad, D.F., Keebler, J.E., Depristo, M.A. et al. (2011). Variation in genome-wide mutation rates within and between human families. *Nat. Genet.* 43: 712–714.
- Cooper, G.M. and Shendure, J. (2011). Needles in stacks of needles: finding disease-causal variants in a wealth of genomic data. *Nat. Rev. Genet.* 12: 628–640.
- Dannemann, M. and Kelso, J. (2017). The contribution of Neanderthals to phenotypic variation in modern humans. *Am. J. Hum. Genet.* 101: 578–589.
- Donnelly, M.P., Paschou, P., Grigorenko, E. et al. (2012). A global view of the OCA2-HERC2 region and pigmentation. *Hum. Genet.* 131: 683–696.
- Duggan, A.T. and Stoneking, M. (2014). Recent developments in the genetic history of East Asia and Oceania. *Curr. Opin. Genet. Dev.* 29: 9–14.
- ENCODE Project Consortium (2012). An integrated encyclopedia of DNA elements in the human genome. *Nature* 489: 57–74.
- Fan, S., Hansen, M.E., Lo, Y., and Tishkoff, S.A. (2016). Going global by adapting local: a review of recent human adaptation. *Science* 354: 54–59.
- Field, Y., Boyle, E.A., Telis, N. et al. (2016). Detection of human adaptation during the past 2000 years. *Science* 354: 760–764.
- Flint, J., Hill, A.V., Bowden, D.K. et al. (1986). High frequencies of alpha-thalassaemia are the result of natural selection by malaria. *Nature* 321: 744–750.
- Fu, W. and Akey, J.M. (2013). Selection and adaptation in the human genome. *Annu. Rev. Genomics Hum. Genet.* 14: 467–489.
- Fu, W., O'Connor, T.D., Jun, G. et al. (2013). Analysis of 6,515 exomes reveals the recent origin of most human protein-coding variants. *Nature* 493: 216–220.
- Fu, Q., Posth, C., Hajdinjak, M. et al. (2016). The genetic history of Ice Age Europe. *Nature* 534: 200–205.
- Fumagalli, M., Moltke, I., Grarup, N. et al. (2015). Greenlandic inuit show genetic signatures of diet and climate adaptation. *Science* 349: 1343–1347.
- Genovese, G., Friedman, D.J., Ross, M.D. et al. (2010). Association of trypanolytic ApoL1 variants with kidney disease in African Americans. *Science* 329: 841–845.
- Grossman, S.R., Shylakhter, I., Karlsson, E.K. et al. (2010). A composite of multiple signals distinguishes causal variants in regions of positive selection. *Science* 327: 883–886.
- Grossman, S.R., Andersen, K.G., Shylakhter, I. et al. (2013). Identifying recent adaptations in large-scale genomic data. *Cell* 152: 703–713.
- Gunther, T. and Jakobsson, M. (2016). Genes mirror migrations and cultures in prehistoric Europe – a population genomic perspective. *Curr. Opin. Genet. Dev.* 41: 115–123.
- Gutenkunst, R.N., Hernandez, R.D., Williamson, S.H., and Bustamante, C.D. (2009). Inferring the joint demographic history of multiple populations from multidimensional SNP frequency data. *PLoS Genet.* 5: e1000695.
- Haasl, R.J. and Payseur, B.A. (2016). Fifteen years of genomewide scans for selection: trends, lessons and unaddressed genetic sources of complication. *Mol. Ecol.* 25: 5–23.
- Hamblin, M.T. and Di Rienzo, A. (2000). Detection of the signature of natural selection in humans: evidence from the Duffy blood group locus. *Am. J. Hum. Genet.* 66: 1669–1679.
- Hamblin, M.T., Thompson, E.E., and Di Rienzo, A. (2002). Complex signatures of natural selection at the Duffy blood group locus. *Am. J. Hum. Genet.* 70: 369–383.

- Hellenthal, G., Busby, G.B., Band, G. et al. (2014). A genetic atlas of human admixture history. *Science* 343: 747–751.
- Holsinger, K.E. and Weir, B.S. (2009). Genetics in geographically structured populations: defining, estimating and interpreting F_{ST} . *Nat. Rev. Genet.* 10: 639–650.
- Hu, H., Petousi, N., Glusman, G. et al. (2017). Evolutionary history of Tibetans inferred from whole-genome sequencing. *PLoS Genet.* 13: e1006675.
- Huerta-Sanchez, E., Jin, X., Asan Bianba, Z. et al. (2014). Altitude adaptation in Tibetans caused by introgression of Denisovan-like DNA. *Nature* 512: 194–197.
- Hughes, A.L. and Yeager, M. (1998). Natural selection at major histocompatibility complex loci of vertebrates. *Annu. Rev. Genet.* 32: 415–435.
- Jablonski, N.G. and Chaplin, G. (2010). Colloquium paper: human skin pigmentation as an adaptation to UV radiation. *Proc. Natl. Acad. Sci. U.S.A.* 107 (Suppl 2): 8962–8968.
- Jeong, C. and Di Rienzo, A. (2014). Adaptations to local environments in modern human populations. *Curr. Opin. Genet. Dev.* 29: 1–8.
- Jorde, L.B. and Wooding, S.P. (2004). Genetic variation, classification, and "race". *Nat. Genet.* 36 (11 Suppl): S28–S33.
- Kayser, M. and De Knijff, P. (2011). Improving human forensics through advances in genetics, genomics and molecular biology. *Nat. Rev. Genet.* 12: 179–192.
- Key, F.M., Teixeira, J.C., De Filippo, C., and Andres, A.M. (2014). Advantageous diversity maintained by balancing selection in humans. *Curr. Opin. Genet. Dev.* 29: 45–51.
- Kwiatkowski, D.P. (2005). How malaria has affected the human genome and what human genetics can teach us about malaria. *Am. J. Hum. Genet.* 77: 171–192.
- Lachance, J. and Tishkoff, S.A. (2013a). SNP ascertainment bias in population genetic analyses: why it is important, and how to correct it. *BioEssays* 35: 780–786.
- Lachance, J. and Tishkoff, S.A. (2013b). Population genomics of human adaptation. *Annu. Rev. Ecol. Evol. Syst.* 44: 123–143.
- Lamason, R.L., Mohideen, M.A., Mest, J.R. et al. (2005). SLC24A5, a putative cation exchanger, affects pigmentation in zebrafish and humans. *Science* 310: 1782–1786.
- Lawson, D.J., Hellenthal, G., Myers, S., and Falush, D. (2012). Inference of population structure using dense haplotype data. *PLoS Genet.* 8: e1002453.
- Lek, M., Karczewski, K.J., Minikel, E.V. et al. (2016). Analysis of protein-coding genetic variation in 60,706 humans. *Nature* 536: 285–291.
- Leslie, S., Winney, B., Hellenthal, G. et al. (2015). The fine-scale genetic structure of the British population. *Nature* 519: 309–314. Wellcome Trust Case Control Consortium
- Li, H. and Durbin, R. (2011). Inference of human population history from individual whole-genome sequences. *Nature* 475: 493–496.
- Liu, Y., Nyunoya, T., Leng, S. et al. (2013). Softwares and methods for estimating genetic ancestry in human populations. *Hum. Genomics* 7: 1.
- Liu, X., Lu, D., Saw, W.Y. et al. (2017). Characterising private and shared signatures of positive selection in 37 Asian populations. *Eur. J. Hum. Genet.* 25: 499–508.
- Lorenzo, F.R., Huff, C., Myllymaki, M. et al. (2014). A genetic mechanism for Tibetan high-altitude adaptation. *Nat. Genet.* 46: 951–956.
- Macinnis, M.J., Koehle, M.S., and Rupert, J.L. (2010). Evidence for a genetic basis for altitude illness: 2010 update. *High Alt. Med. Biol.* 11: 349–368.
- Malaspinas, A.S., Westaway, M.C., Muller, C. et al. (2016). A genomic history of Aboriginal Australia. *Nature* 538: 207–214.
- Mallick, S., Li, H., Lipson, M. et al. (2016). The Simons Genome Diversity Project: 300 genomes from 142 diverse populations. *Nature* 538: 201–206.
- Manolio, T.A. (2013). Bringing genome-wide association findings into clinical use. *Nat. Rev. Genet.* 14: 549–558.
- Marigorta, U.M. and Navarro, A. (2013). High trans-ethnic replicability of GWAS results implies common causal variants. *PLoS Genet.* 9: e1003566.
- Mathieson, I., Lazaridis, I., Rohland, N. et al. (2015). Genome-wide patterns of selection in 230 ancient Eurasians. *Nature* 528: 499–503.

- Moorjani, P., Patterson, N., Hirschhorn, J.N. et al. (2011). The history of African gene flow into Southern Europeans, Levantines, and Jews. *PLoS Genet.* 7: e1001373.
- Moorjani, P., Gao, Z., and Przeworski, M. (2016). Human germline mutation and the erratic evolutionary clock. *PLoS Biol.* 14: e2000744.
- Nachman, M.W. and Crowell, S.L. (2000). Estimate of the mutation rate per nucleotide in humans. *Genetics* 156: 297–304.
- Nielsen, R., Akey, J.M., Jakobsson, M. et al. (2017). Tracing the peopling of the world through genomics. *Nature* 541: 302–310.
- Norton, H.L., Kittles, R.A., Parra, E. et al. (2007). Genetic evidence for the convergent evolution of light skin in Europeans and East Asians. *Mol. Biol. Evol.* 24: 710–722.
- Novembre, J. and Peter, B.M. (2016). Recent advances in the study of fine-scale population structure in humans. *Curr. Opin. Genet. Dev.* 41: 98–105.
- Novembre, J. and Ramachandran, S. (2011). Perspectives on human population structure at the cusp of the sequencing era. *Annu. Rev. Genomics Hum. Genet.* 12: 245–274.
- Novembre, J. and Stephens, M. (2008). Interpreting principal component analyses of spatial population genetic variation. *Nat. Genet.* 40: 646–649.
- Ostrer, H. and Skorecki, K. (2013). The population genetics of the Jewish people. *Hum. Genet.* 132: 119–127.
- Patterson, N., Moorjani, P., Luo, Y. et al. (2012). Ancient admixture in human history. *Genetics* 192: 1065–1093.
- Perry, G.H., Dominy, N.J., Claw, K.G. et al. (2007). Diet and the evolution of human amylase gene copy number variation. *Nat. Genet.* 39: 1256–1260.
- Pickrell, J.K. and Pritchard, J.K. (2012). Inference of population splits and mixtures from genome-wide allele frequency data. *PLoS Genet.* 8: e1002967.
- Piel, F.B., Patil, A.P., Howes, R.E. et al. (2010). Global distribution of the sickle cell gene and geographical confirmation of the malaria hypothesis. *Nat. Commun.* 1: 104.
- Price, A.L., Patterson, N.J., Plenge, R.M. et al. (2006). Principal components analysis corrects for stratification in genome-wide association studies. *Nat. Genet.* 38: 904–909.
- Pritchard, J.K., Stephens, M., and Donnelly, P. (2000). Inference of population structure using multilocus genotype data. *Genetics* 155: 945–959.
- Pritchard, J.K., Pickrell, J.K., and Coop, G. (2010). The genetics of human adaptation: hard sweeps, soft sweeps, and polygenic adaptation. *Curr. Biol.* 20: R208–R215.
- Purcell, S., Neale, B., Todd-Brown, K. et al. (2007). PLINK: a tool set for whole-genome association and population-based linkage analyses. *Am. J. Hum. Genet.* 81: 559–575.
- Quintana-Murci, L. (2016). Understanding rare and common diseases in the context of human evolution. *Genome Biol.* 17: 225.
- Racimo, F., Sankararaman, S., Nielsen, R., and Huerta-Sanchez, E. (2015). Evidence for archaic adaptive introgression in humans. *Nat. Rev. Genet.* 16: 359–371.
- Raj, A., Stephens, M., and Pritchard, J.K. (2014). fastSTRUCTURE: variational inference of population structure in large SNP data sets. *Genetics* 197: 573–589.
- Ramachandran, S., Deshpande, O., Roseman, C.C. et al. (2005). Support from the relationship of genetic and geographic distance in human populations for a serial founder effect originating in Africa. *Proc. Natl. Acad. Sci. U.S.A.* 102: 15942–15947.
- Rees, D.C., Williams, T.N., and Gladwin, M.T. (2010). Sickle-cell disease. *Lancet* 376: 2018–2031.
- Reich, D., Thangaraj, K., Patterson, N. et al. (2009). Reconstructing Indian population history. *Nature* 461: 489–494.
- Reich, D., Green, R.E., Kircher, M. et al. (2010). Genetic history of an archaic hominin group from Denisova Cave in Siberia. *Nature* 468: 1053–1060.
- Reiff, S.B. and Striepen, B. (2009). Malaria: the gatekeeper revealed. *Nature* 459: 918–919.
- Roach, J.C., Glusman, G., Smit, A.F. et al. (2010). Analysis of genetic inheritance in a family quartet by whole-genome sequencing. *Science* 328: 636–639.
- Rosenberg, N.A. and Nordborg, M. (2002). Genealogical trees, coalescent theory and the analysis of genetic polymorphisms. *Nat. Rev. Genet.* 3: 380–390.

- Rosenberg, N.A., Mahajan, S., Ramachandran, S. et al. (2005). Clines, clusters, and the effect of study design on the inference of human population structure. *PLoS Genet.* 1: e70.
- Rosenberg, N.A., Huang, L., Jewett, E.M. et al. (2010). Genome-wide association studies in diverse populations. *Nat. Rev. Genet.* 11: 356–366.
- Royal, C.D., Novembre, J., Fullerton, S.M. et al. (2010). Inferring genetic ancestry: opportunities, challenges, and implications. *Am. J. Hum. Genet.* 86: 661–673.
- Sabeti, P.C., Reich, D.E., Higgins, J.M. et al. (2002). Detecting recent positive selection in the human genome from haplotype structure. *Nature* 419: 832–837.
- Sabeti, P.C., Varilly, P., Fry, B. et al. (2007). Genome-wide detection and characterization of positive selection in human populations. *Nature* 449: 913–918.
- Sankararaman, S., Mallick, S., Dannemann, M. et al. (2014). The genomic landscape of Neanderthal ancestry in present-day humans. *Nature* 507 (7492): 354–357.
- Savage, S.A., Gerstenblith, M.R., Goldstein, A.M. et al. (2008). Nucleotide diversity and population differentiation of the melanocortin 1 receptor gene, MC1R. *BMC Genet.* 9: 31.
- Scheinfeldt, L.B. and Tishkoff, S.A. (2013). Recent human adaptation: genomic approaches, interpretation and insights. *Nat. Rev. Genet.* 14: 692–702.
- Schiffels, S. and Durbin, R. (2014). Inferring human population size and separation history from multiple genome sequences. *Nat. Genet.* 46: 919–925.
- Schraiber, J.G. and Akey, J.M. (2015). Methods and models for unravelling human evolutionary history. *Nat. Rev. Genet.* 16: 727–740.
- Segurel, L., Wyman, M.J., and Przeworski, M. (2014). Determinants of mutation rate variation in the human germline. *Annu. Rev. Genomics Hum. Genet.* 15: 47–70.
- Shendure, J. and Akey, J.M. (2015). The origins, determinants, and consequences of human mutations. *Science* 349: 1478–1483.
- Sherry, S.T. and Batzer, M.A. (1997). Modeling human evolution – to tree or not to tree? *Genome Res.* 7: 947–949.
- Simonson, T.S., Yang, Y., Huff, C.D. et al. (2010). Genetic evidence for high-altitude adaptation in Tibet. *Science* 329: 72–75.
- Simonson, T.S., Huff, C.D., Witherspoon, D.J. et al. (2015). Adaptive genetic changes related to haemoglobin concentration in native high-altitude Tibetans. *Exp. Physiol.* 100: 1263–1268.
- Skoglund, P. and Reich, D. (2016). A genomic view of the peopling of the Americas. *Curr. Opin. Genet. Dev.* 41: 27–35.
- Strauss, K.A. and Puffenberger, E.G. (2009). Genetics, medicine, and the plain people. *Annu. Rev. Genomics Hum. Genet.* 10: 513–536.
- Sturm, R.A. and Duffy, D.L. (2012). Human pigmentation genes under environmental selection. *Genome Biol.* 13: 248.
- Sudmant, P.H., Rausch, T., Gardner, E.J. et al. (2015a). An integrated map of structural variation in 2,504 human genomes. *Nature* 526: 75–81.
- Sudmant, P.H., Mallick, S., Nelson, B.J. et al. (2015b). Global diversity, population stratification, and selection of human copy number variation. *Science* <https://doi.org/10.1126/science.aab3761>.
- Tang, H., Peng, J., Wang, P., and Risch, N.J. (2005). Estimation of individual admixture: analytical and study design considerations. *Genet. Epidemiol.* 28: 289–301.
- Tennessen, J.A., Bigham, A.W., O’Connor, T.D. et al. (2012). Evolution and functional impact of rare coding variation from deep sequencing of human exomes. *Science* 337 (6090): 64–69.
- Teshima, K.M., Coop, G., and Przeworski, M. (2006). How reliable are empirical genomic scans for selective sweeps? *Genome Res.* 16: 702–712.
- Tishkoff, S.A., Varkonyi, R., Cahinhinan, N. et al. (2001). Haplotype diversity and linkage disequilibrium at human G6PD: recent origin of alleles that confer malarial resistance. *Science* 293: 455–462.
- Tishkoff, S.A., Reed, F.A., Ranciaro, A. et al. (2007). Convergent adaptation of human lactase persistence in Africa and Europe. *Nat. Genet.* 39: 31–40.

- UK10K Consortium, Walter, K., Min, J.L. et al. (2015). The UK10K Project identifies rare variants in health and disease. *Nature* 526: 82–90.
- Veeramah, K.R. and Hammer, M.F. (2014). The impact of whole-genome sequencing on the reconstruction of human population history. *Nat. Rev. Genet.* 15: 149–162.
- Visscher, P.M., Wray, N.R., Zhang, Q. et al. (2017). 10 years of GWAS discovery: biology, function, and translation. *Am. J. Hum. Genet.* 101: 5–22.
- Vitti, J.J., Grossman, S.R., and Sabeti, P.C. (2013). Detecting natural selection in genomic data. *Annu. Rev. Genet.* 47: 97–120.
- Voight, B.F., Kudaravalli, S., Wen, X., and Pritchard, J.K. (2006). A map of recent positive selection in the human genome. *PLoS Biol.* 4: e72.
- Volkman, S.K., Barry, A.E., Lyons, E.J. et al. (2001). Recent origin of *Plasmodium falciparum* from a single progenitor. *Science* 293: 482–484.
- Wang, C., Zöllner, S., and Rosenberg, N.A. (2012). A quantitative comparison of the similarity between genes and geography in worldwide human populations. *PLoS Genet.* 8: e1002886.
- Witherspoon, D.J., Wooding, S., Rogers, A.R. et al. (2007). Genetic similarities within and between human populations. *Genetics* 176: 351–359.
- Xing, J., Watkins, W.S., Shlien, A. et al. (2010). Toward a more uniform sampling of human genetic diversity: a survey of worldwide populations by high-density genotyping. *Genomics* 96: 199–210.
- Yi, X., Liang, Y., Huerta-Sanchez, E. et al. (2010). Sequencing of 50 human exomes reveals adaptation to high altitude. *Science* 329: 75–78.

16

Metagenomics and Microbial Community Analysis

Robert G. Beiko

Introduction

The human microbiome, defined by Joshua Lederberg as “...the ecological community of commensal, symbiotic, and pathogenic microorganisms that literally share our body space,” is of central importance to human health. Environmental microbiomes are equally important as they support the basic biological processes upon which all life depends, and their response to changing climate will fundamentally influence the health of the biosphere. We can now investigate the composition and function of the microbiome in depth using techniques such as metagenomics, where environmental DNA sequencing is used to provide an in-depth cross-section of the taxonomic and functional diversity of a metagenome sample. Metagenomics and other techniques have driven a rapid growth in the scope of exploratory and experimental studies performed on the microbiome. While metagenomic data analysis makes use of classical bioinformatic techniques including sequence alignment and homology assessment, the high level of biological diversity in most samples (often >100 named species) and fragmentary nature of the sequence data present unique challenges that require the development of specialized bioinformatic techniques.

Metagenomic sequence analysis and related approaches can provide a detailed, if incomplete, profile of the microbiome. Modern DNA sequencing platforms can generate tens of millions of short (150–250 nt in length) DNA sequences that represent a sample of the total DNA present in a microbial community. Starting with these sequence reads, the challenge of metagenomic data analysis is to infer community structure and function in a given sample and to enable comparisons across multiple samples. With taxonomic and functional information in hand, researchers can ask detailed questions about microbial communities, such as: “Are there consistent microbial signatures in chronic conditions such as Crohn’s disease and type II diabetes?,” “What is the microbial carbohydrate catabolic potential in cattle rumen?,” “How is the human microbiome influenced by diet and how does it impact human health?,” “What constitutes ‘normal’ seasonal variation in freshwater microbial communities?,” and “How is this seasonal variation influenced by rare disturbance events?”

Many computational steps are needed to span the gap between raw sequence reads and robust ecological inferences. Although many excellent algorithms and software tools have been developed for microbiome analysis, many methods make assumptions that may be at odds with the actual structure and function of the community. A widely cited example is the problem of compositionality in normalized taxonomic abundance profiles. For instance, representing sequence counts as microbial population proportions can induce false correlations among taxa and incorrect inferences of community structure. To provide an overview of metagenomic data analysis, this chapter will focus on three central challenges: (i) making sense of fragmentary and incomplete DNA sequence data, (ii) using appropriate representations of community diversity and function, and (iii) finding appropriate techniques to summarize and compare microbiome samples.

Although prokaryotic organisms (i.e. bacteria and archaea) are often the focus of microbiome studies, other entities including viruses, single-celled eukaryotes, and even small multicellular organisms such as nematodes can be considered as part of the microbiome. However, the choice of experimental technique strongly influences the type of information that is recovered. For example, marker-gene analyses that amplify a specific region of the genome via the polymerase chain reaction (PCR) often target only prokaryotes, and, depending on the choice of primers, may return information only for bacteria. Assessing viruses requires a different set of techniques including selective filtration, and computational approaches that depend on a single, universal marker gene cannot be used. The choice of target organisms may be explicitly hypothesis driven, or implicit owing to the choice of assessment technique.

Why Study the Microbiome?

In a study published in 2017, researchers examined the effect of microbiota transfer therapy on symptoms associated with autism spectrum disorder (ASD). The trial introduced a “healthy” set of gut microbes into a set of children with ASD over the span of 7–8 weeks. The findings of the study were remarkable: not only did the severity of gastrointestinal symptoms drop by over 80%, but clinical symptoms associated with autism showed marked and persistent improvement (Kang et al. 2017). How can microbial therapy have such a profound effect on health and cognitive status? The key lies in the metabolic and signaling interactions that link the gut microbiota and the host. In the case of autism, different types of bacteria have been associated positively and negatively with ASD, although no precise microbial “signature” has yet been identified. The mechanisms that drive host–microbe interactions in ASD are difficult to assess and work is ongoing, but immune dysregulation induced by the microbiome is likely to play a key role (Vuong and Hsiao 2017).

A key element to assessing the impact of interventions such as microbial therapy on the patient is an assessment of changes in the microbial community that are induced by the intervention. To do this, it is necessary to perform “before and after” assessments of the microbiota. How can this be done? Culturing the complete repertoire of bacteria is impractical and, in many cases, impossible because of the large number of species and the recalcitrance of many species to be grown in the laboratory. The most widely used approach is to collect molecular data using high-throughput DNA sequencing approaches, then use bioinformatic techniques and reference databases to connect the observed information (such as DNA sequences) with information about the taxonomic diversity and molecular functions present in a sample. In the case of the autism study outlined above, the authors used DNA sequences as proxies for microbial biodiversity, and were able to correlate the abundance of taxa such as *Bifidobacterium* and *Prevotella* with patient improvements in a number of standardized assessments for symptoms of autism.

Hypothesis-driven experimental and clinical studies can provide detailed information about the relationship between changes in the microbiome and changes in, for example, the health of a human host. Even surveys of the microbiome from a set of human subjects or environmental samples can be highly informative. Large-scale microbiome mapping projects, including the first Human Microbiome Project (Turnbaugh et al. 2007; Huttenhower et al. 2012), focused their efforts on constructing a baseline microbiome found in nominally healthy human subjects. Many studies have compared the microbiome of individuals in different countries, different geographic areas within the same country, or even different buildings and transit centers within a single city. Metagenomic analysis has also been used to identify and profile specific functional genes within the human microbiome, including genes for vitamin and short-chain fatty acid synthesis, as well as genes that confer antimicrobial resistance.

Environmental microbiome studies have expanded our knowledge of the microorganisms that underpin nutrient cycling and sustain most ecological interactions. Early investigations

of the ocean microbiome revealed the presence of SAR11, which is now known to be the most abundant group of microorganisms in the ocean. Metagenomic analysis revealed a significant amount of variation and potential environmental filtering of SAR11 types, suggesting different ecological interactions and different impacts on nutrient cycling depending on the geographic location (Brown et al. 2012; Giovannoni 2017), and latitudinal gradients of biodiversity and function. Metagenomic analysis of the Gulf of Mexico after the 2010 Deepwater Horizon oil spill revealed an increased presence of organisms such as *Oceanospirillum* that were capable of oil degradation (Lu et al. 2012). Although this increase suggests a potential for the use of *Oceanospirillum* in bioremediation, its impact in the aftermath of the Deepwater Horizon oil spill is uncertain. Other large-scale environmental surveys have also been completed, such as the Tara Oceans initiative (Sunagawa et al. 2015) and the Earth Microbiome Project (Thompson et al. 2017). In addition to the discovery of novel taxonomic groups and ecological associations, such surveys provide a baseline and a reference database for future studies. As the Earth's biodiversity suffers increasing pressure as a result of environmental damage and climate change, understanding the environmental microbiome will be of central importance in both assessing and anticipating the impacts of change, and developing mitigation strategies.

The Origins of Microbiome Analysis

A pioneering analysis in 1985 profiled genetic sequence data directly from the environment (Stahl et al. 1985). Using direct sequencing of 5S ribosomal RNA (rRNA), the authors were able to identify relatives of cultured thermophilic organisms and describe the structure of the microbial community in Octopus Spring at Yellowstone National Park by mapping the recovered sequence into a larger phylogenetic tree. This thermal hot spring, with a slightly alkaline pH and a temperature of 91 °C, was found to have a thriving microbial community with members living at higher temperature and pH than their previously characterized distant relatives. In 1991 the first study based on the 16S rRNA gene (henceforth “16S” or “16S gene”) was published, which was used to describe community structure in the Sargasso Sea (the deep blue sea surrounding Bermuda). With the level of resolution afforded by the 16S gene, the authors were able to discover the highly diverse and abundant SAR11 cluster of organisms (Giovannoni et al. 1990). By the mid-1990s, the pace of 16S surveys was accelerating, driven in part by decreases in the cost of Sanger sequencing and the rapid increase in available 16S data. As a result, the Ribosomal Database Project (RDP; Cole et al. 2014) saw a massive increase in its sequence collection, from just 471 to over 100 000 ribosomal sequences in the 10 year period between 1992 and 2001.

Although metagenomics has become nearly synonymous with DNA sequencing in the twenty-first century, the first use of the term was by Handelsman et al. (1998), who used functional cloning vectors to express genes isolated directly from the environment. The term “metagenomics” refers to the study of multiple genomes at a time in a manner that does not depend on bacterial isolation and culture. However, the term grew to encompass other environmental characterization approaches, particularly direct DNA sequencing from the environment, which was the centerpiece of landmark papers that focused on characterizing the microbiome of the Sargasso Sea and acid-mine drainage environments (Tyson et al. 2004; Venter et al. 2004). Despite the utility of DNA sequencing in characterizing microbial communities, the presence of genes in a sample does not guarantee that they are actively contributing to biochemical processes in the corresponding system. DNA surveys based on marker genes and metagenome samples do not differentiate the expression of genes under different conditions, or the metabolic consequences of gene expression. This limitation has spawned an expanded set of “meta-omic” or “multi-omic” techniques, so named because they span multiple methods of assessment, including metatranscriptomics, metaproteomics,

and meta-metabolomics that target community RNA, protein, and metabolite composition, respectively. These approaches can distinguish samples that do not show substantial differences in DNA content. However, these techniques are more expensive and sample preparation and storage are typically more complicated. These emerging methods hold great promise, but, given the widespread use of DNA-based approaches, the majority of this chapter will focus on marker-gene and metagenomic analysis.

Metagenomic Workflow

There are many variations in the analysis of metagenomic and related types of data, but there are strong similarities in the experimental and analytical pipelines used for all such studies (Figure 16.1).

- Sample collection is performed according to a standard collection protocol. This protocol may involve filtering by size, core sampling, swabbing, and the like. An important consideration is the choice of conditions used to store the DNA. For instance, leaving samples at room temperature for a substantial length of time can negatively impact the composition of the microbial community along with the taxonomic and functional distribution of the generated sequences (Choo et al. 2015). Samples prepared for other types of approaches (e.g. metatranscriptomics) must be treated with extra care. For example, samples that will be characterized using mass spectrometry must not be frozen prior to extraction.
- The technique used for DNA sequence extraction typically depends on the source habitat for reasons of chemistry and physical consistency of samples. For example, soil samples typically contain humic acids – compounds that interfere with enzymes used to prepare and sequence DNA – so techniques such as DNA purification must be performed prior to DNA sequencing.
- Preparation for DNA sequencing is then performed by constructing a sequence “library.” A sequence library is a collection of specially prepared DNA fragments derived from the genome or genomes being sequenced. The library preparation step may involve DNA shearing, amplification via PCR, the addition of adapters to facilitate sequencing, and the addition of short, distinctive DNA sequences (Hamady et al. 2008) to distinguish multiple samples that are being sequenced in the same run. The preparation protocol is usually specific to the DNA sequencing platform that will be used.

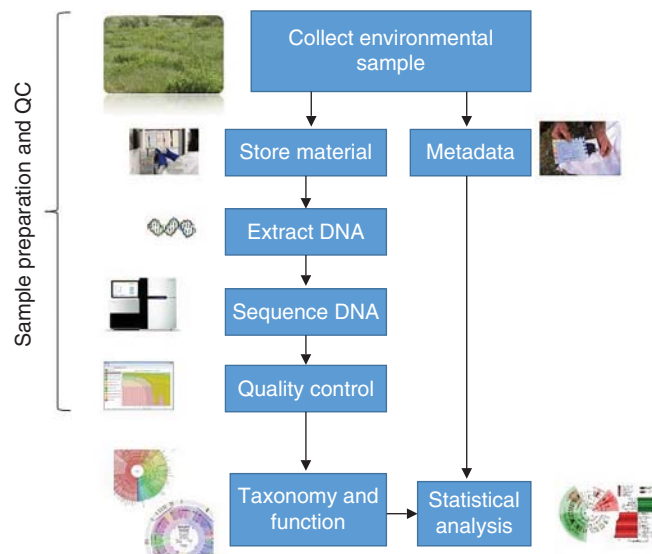


Figure 16.1 General workflow for DNA-based microbiome analysis.

General Considerations in Marker-Gene and Metagenomic Data Analysis

Refinements to the general workflow described above can have significant impacts on the downstream bioinformatic analyses that need to be performed.

The choice of DNA sequencing platform can strongly influence the recovery of information about the microbiome. DNA sequencing technology is changing rapidly, and the selection of a sequencing platform for a given analysis will depend on a combination of factors including cost, expected error rate, expected error profile, read length, and availability. As the sequencing cost per nucleotide base continues to drop and new technologies emerge, the best choice of platform for a given project will change. In general, diversity studies benefit from longer read lengths. Marker-gene studies are often based on short fragments of a gene, but sequencing longer fragments or the entire gene provides more resolution for taxonomic classification. In shotgun metagenome studies, the determination of which DNA fragments came from which genomes is greatly assisted by the availability of reads that are >1000 nucleotides in length. Such reads are more likely to span regions that are difficult to assemble, such as repetitive, low-complexity regions or paralogous genes. This facilitates the assembly of larger contigs.

Several sequencing platforms currently account for nearly all environmental DNA studies. The Illumina series of DNA sequencing machines produce a very large number of sequence reads at relatively low cost, but sequence reads are short – typically only 150–300 nt in length. The sequencing error rate of Illumina machines is approximately 0.1%. Given their popularity, short-read Illumina sequence data will be the focus of much of this chapter, although other techniques will be mentioned as appropriate. The Pacific Biosciences (PacBio) RS II produces much longer reads, with a median read length typically between 5000 and 10 000 nt. While this can be an advantage for microbial community analysis, the sequencing error rate (>10%) is very high. It is common in many applications to use long PacBio reads to create an approximate scaffold of metagenomic contigs, then do Illumina sequencing to correct errors to the greatest extent possible. Nanopore sequencing is relatively new and is beginning to see applications in many areas of investigation. Like PacBio, the Oxford Nanopore MinION can produce long reads, although it also suffers from a relatively high error rate. The Nanopore approach is the most costly per nucleotide of sequence, but its current key advantages are portability and the ability to do “real-time” sequencing.

Data quality is a significant concern in metagenomic and marker-gene studies. As mentioned above, sequencing errors can present significant challenges to analysis. All DNA sequencing platforms produce errors, but different platforms generate different types of errors (e.g. sequence substitutions versus insertions). It is therefore necessary to filter sequence reads based on quality metrics, which are often represented in a FASTQ-formatted file (Cock et al. 2009). Phred scores are a widely used measure of sequence quality, and express the probability of a base being called incorrectly. FASTQ files augment the typical sequence and header information with a base-by-base representation of Phred scores. FastQC (www.bioinformatics.babraham.ac.uk/projects/fastqc) is a widely used tool to summarize sequence reads from a sample based on their Phred scores, and perform trimming of low-quality sequence regions and removal of low-quality reads (Figure 16.2).

Typically, reads with an average quality score below a given threshold will be removed, as will sections of reads with low-quality base calls. Although different quality thresholds can be used, a common approach is to truncate reads after encountering a nucleotide with associated quality less than a given threshold, or to discard a read entirely. Chimeric sequences can also arise during the process of PCR; since these are derived from parts of two different sequences, they are very misleading from a taxonomic perspective and should be removed.

The computational cost of analysis is also important. Given that many metagenomic datasets contain a million or more sequence reads, applications whose run time or memory usage scales quadratically, or worse, with input dataset size will not be viable unless the datasets are very small or have been aggressively filtered prior to a detailed analysis. A simple example would

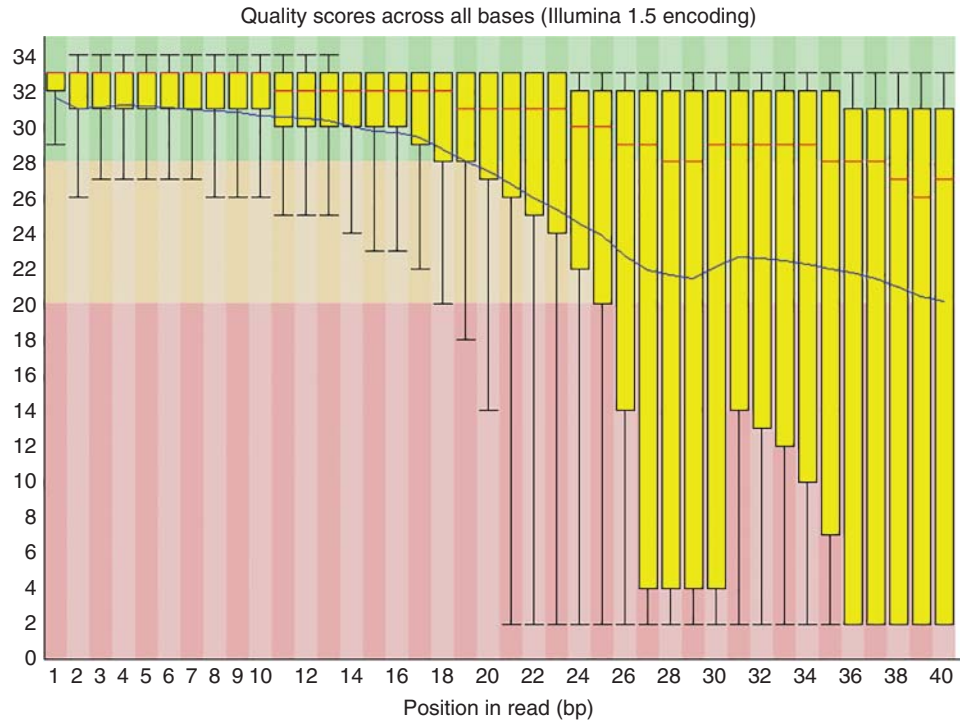


Figure 16.2 FastQC summary of DNA sequence read quality for an Illumina sequencing run. For each read position (horizontal axis), the distribution of quality scores (vertical axis) is shown. Although the first few read positions have good scores on average (green band), quality score distributions increase rapidly, with many sites showing poor quality across all reads (red band). Source: www.bioinformatics.babraham.ac.uk/projects/fastqc/bad_sequence_fastqc.html.

be an all-versus-all homology search. In this example, a dataset containing n sequences would require approximately $n \times n = n^2$ comparisons, but a dataset twice as large would require $2n \times 2n = 4n^2$ comparisons. With metagenomic datasets comprising hundreds of millions of reads and reference databases comprising tens of thousands of sequenced genomes, these scaling properties are prohibitive and efficient alternatives are required. Sequence alignment is an area where scaling performance has been intensively studied; see Baichoo and Ouzounis (2017) for a review.

The two primary classes of sequence-classification approaches used in microbiome analysis are sequence searching/alignment and compositional similarity. Sequence searching and alignment can be used to identify homology and the degree of similarity between query sequences and sequences in a reference database (see Chapter 3). Compositional similarity is a method in which sequences are decomposed into summary vectors wherein each vector element describes one attribute of the sequence. By far the most common such representation involves k -mers (described in greater detail in Taxonomic Assignment and Profiling), where a sequence is decomposed into all its constituent words of length k . While k -mers sacrifice information about the positional context of words within the larger sequence, they can be orders of magnitude faster to compute than sequence alignment approaches. Sequence searching/alignment and compositional similarity approaches have been adapted from methods developed before the advent of metagenomic sequencing, but refinements to these methods have been necessary to address the unique aspects of metagenomic datasets.

Another important consideration in metagenomic data analysis is the completeness and reliability of reference databases. Databases of *16S* genes now contain in excess of 2 million sequences with assigned taxonomic information, but environmental *16S* surveys inevitably yield sequences with very low levels of similarity to any sequences in the reference database. What should be done with these mystery sequences? One option is to use an unsupervised

approach that associates environmental sequences with one another based on sequence identity or some other criterion, rather than via comparison with a reference database. Another option is to report information at some higher level; for example, a given sequence might be confidently classified to phylum Proteobacteria, but not to any more specific level. In this case, the sequence could be included in a high-level taxonomic summary but excluded from other ranks. Reference taxonomic databases also demonstrate the limitations of prokaryotic taxonomy, as sequences associated with different taxonomic groups are often commingled in *16S* trees (see Chapter 9).

Similar limitations apply in the assignment of functional annotations to metagenomic reads. Functional assignment via homology (using BLASTX or other sequence-searching algorithms outlined below; see also Chapter 3) is common practice, but reference databases contain many genes of unknown function. Even in *Escherichia coli* strain K-12 MG1655, the most intensively studied of all microorganisms, many predicted genes are identified as “hypothetical” or “putative.” Predicted genes in many metagenomic sequence datasets often match genes of no known function and must, therefore, be discarded or binned as “other” in functional summaries.

Another important consideration is the *ecological relevance* of observed microbial/sequence diversity collected from a given sample site. The Baas-Becking hypothesis that “Everything is everywhere, but, the environment selects” is true for many microorganisms, as they are highly mobile and not limited by barriers to migration (Baas-Becking 1934). Nearby habitats are highly connected, as is clearly illustrated by the transmission of pathogens from host to host, as well as the recolonization of nearby habitats after an environmental disturbance. While fecal samples are often used as proxies for the human gut microbiome, targeted studies have demonstrated that different parts of the gut can harbor different types of microorganisms, and a fecal sample may represent a mixture of organisms from different sub-habitats (Stearns et al. 2011). Migration and habitat connectivity can lead to microbial communities which contain organisms that are ecologically irrelevant. Deep sequencing of ocean habitats (Caporaso et al. 2012) has revealed the presence of many sequences at extremely low levels of abundance. As a result, the organisms represented by these sequences may, in many cases, have a minimal impact on the surrounding environment. However, it cannot be assumed that rarity implies irrelevance, as rare groups can provide essential metabolic processes, and some rare taxa show occasional spikes in abundance, a property known as *conditional rarity* (Shade et al. 2014). Repeated sampling and repeated experiments can potentially distinguish rare from irrelevant taxa, but, in any case, the researcher must be careful in making assumptions about the ecological relevance of observed taxa and functions.

Marker Genes

A *marker gene* is any DNA sequence with some expressed function (typically a structural RNA or protein) that is present in all members of a taxonomic group of interest, and sufficiently variable to distinguish different members of the group. The *16S* gene is a marker gene that dominates general surveys of bacterial and/or archaeal diversity, for several reasons. First, as a key structural component of the ribosome, 16S rRNA is present in all prokaryotes. Second, since the *16S* gene has long served as the basis for early molecular phylogenetic studies, it has been extensively characterized in many microorganisms and serves as the most comprehensive genetic biodiversity resource available. Third, the *16S* gene contains highly conserved regions that serve as useful sites for PCR priming across a wide swath of biodiversity. Fourth, conserved regions flank nine variable regions that are specific enough to resolve many groups to the genus, species, or strain level (Figure 16.3). Fifth, since the *16S* gene does not encode a protein, it does not exhibit the usual pattern of codon degeneracy, which would complicate primer design.

Consequently, using a relatively simple protocol involving the isolation and sequencing of *16S* genes, it is possible to characterize the majority of bacterial diversity in a sample. This

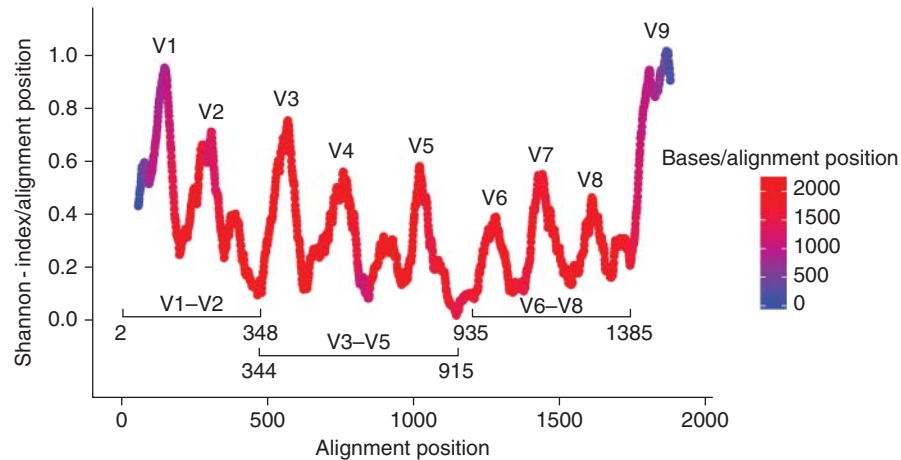


Figure 16.3 Primary structure and variable regions of the *16S* ribosomal RNA gene. Hypervariable regions with high entropy can distinguish different lineages of bacteria, while low-entropy regions are suitable for polymerase chain reaction primer design. Colors indicate the number of times a given homologous site is observed in archaeal sequences in RIM-DB (Seedorf et al. 2014). Source: doi.org/10.7717/peerj.494/fig-2.

strategy has been successfully implemented in thousands of environmental DNA surveys. However, when designing a *16S* experiment, it is important to keep in mind several limitations of the approach. First, multi-copy genes can lead to over-representation of some bacteria in an amplified sample, requiring stoichiometric corrections (Kembel et al. 2012). In rare cases, multiple copies may differ by up to 10% in their nucleotide sequence, making correct species assignment difficult. Second, when designing an amplicon experiment based on short reads, it is important to choose specific primer pairs that target a subset of all variable regions. Primer bias may lead to over-representation of certain genes at the expense of others, and each choice of primer pairs can miss certain genes entirely. Studies performed with different variable regions cannot be compared reliably. Third, the *16S* gene cannot always resolve very close relationships within species or isolates, as any observed variation may be due to sequencing errors.

In spite of these limitations, the *16S* gene remains the most widely used target for metagenomic studies and can generate robust results that are at least internally consistent within a study. If the drawbacks of *16S* are unacceptable for a given study, a common approach is to use alternative markers or to make taxonomic inferences directly from metagenomic data as described in Metagenomic Data Analysis. Examples of markers that are useful across a wide range of taxa include the chaperone gene *cpn60*, which is present in single copy across a broad range of microorganisms, and *rpoB'*, which encodes a subunit of the RNA polymerase holoenzyme, which is useful for some taxonomic groups such as halophilic archaea. Lineage-specific marker genes can provide better resolution at the species and strain level: for example, within the proposed species *Candidatus* “*Accumulibacter phosphatis*” the polyphosphate kinase (*ppk*) gene is often used to resolve ecologically distinct sublineages within the group. The intergenic transcribed region (ITS) between the *16S* and *23S* genes (Box 16.1) is highly variable because of diminished selection pressures, and can also be used for profiling of closely related taxa within a sample. Alternative markers are far less extensively characterized than the *16S* gene, but are nonetheless sufficient to resolve diversity at the desired level in many cases.

Box 16.1 Ribosomal RNA Genes

The ribosome is the protein synthesis center of the cell and is essential to all living organisms. Its structure is very complex, and in prokaryotes the ribosome comprises three RNA molecules (termed ribosomal RNA or rRNA) and approximately 50 different proteins. The prokaryotic rRNAs are referred to by their Svedberg sedimentation constant,

with 5S (~120 nt in length), 16S (~1500 nt), and 23S (~2900 nt) subunits. In addition to their universal distribution across the tree of life, rRNA genes have an advantage over protein-coding genes as their diversity is not influenced by redundancy in the genetic code since they are not translated into protein. Instead, mutation and substitution rates in 16S are governed by structural interactions with varying degrees of conservation. Among rRNA genes, 5S is often viewed as too short and lacking sufficient variational information to support broad phylogenetic surveys, although its length was an advantage in early studies. The 23S gene has considerable potential (Hunt et al. 2006) and is indexed in databases such as the Ribosomal Database Project and SILVA, but it has been far less intensively studied and is much more poorly represented in reference databases, validated polymerase chain reaction primer pairs, and protocols. Between the 16S and 23S lies the intergenic (or internal) transcribed spacer (ITS) region, which is present in all prokaryotes but evolves much more quickly because of diminished selective pressures. This region, which can contain short transfer RNA-encoding genes, is very useful for differentiating closely related organisms whose 16S rRNA genes may be identical. For example, Rocap et al. (2002) used ITS sequences to differentiate the closely related but highly ecologically diverse cyanobacterial genera *Prochlorococcus* and *Synechococcus*.

Eukaryotic organisms have the homologous 18S rRNA gene, which serves the same function in the ribosome as the 16S gene and is also a useful taxonomic marker. The 18S gene shares many of the advantages and disadvantages of the 16S gene. However, it is difficult to target 18S analysis to the microbiome specifically, since many microbial eukaryotes share greater similarity with their closest multicellular relatives than they do with one another. In some cases, these microbial eukaryotes can also exhibit highly divergent 18S sequences that make the generation of universal primers difficult. Another problem linked with 18S microbiome analysis is that many habitats, including the human gut, are flooded with host DNA that can dominate the results of a DNA sequencing run. Since these host DNA sequences are often not of interest, they represent wasted sequencing effort that impedes the recovery of adequate microbiome 18S information. Techniques such as the use of blocking primers that specifically interfere with the amplification of host DNA are often used.

Although many software tools have been developed to execute different steps in the bioinformatic analysis of 16S sequences, two software packages, QIIME (Caporaso et al. 2010) and mothur (Schloss et al. 2009), implement standardized pipelines that have been used in the majority of 16S analyses. Variants that address important limitations and assumptions of the basic workflow have been developed; in some cases, these variants are gaining widespread use.

Quality Control

PCR chimera checking is done using software such as UCHIME (Edgar et al. 2011). The clearest signal of a potential chimeric sequence is when different regions of a sampled sequence have strong matches to different sequences in a reference database, or in the sample. Rather than attempt to reconcile each part of the sampled sequence, it is simply discarded from the analysis.

Grouping of Similar Sequences

Once quality control has been performed, sequences must be assigned to groups that will serve as indivisible “units” in a downstream analysis. The most straightforward approach is to treat each group of unique sequences separately, but this strategy has two key limitations. First, quality control cannot screen out all sequencing errors, and the filtered dataset will still contain sequences that differ from true environmental sequences. Second, an environmental survey can generate tens of thousands of unique sequences or more, which may pose a barrier

to downstream analyses whose efficiency decays rapidly with increasing numbers of units. Therefore, it is common to use a clustering strategy that groups sequences into larger units based on some measure of similarity. Alternatively, newer approaches aim to build clusters that are flexibly defined based on the distribution of similar sequences in the sample, rather than on a universal similarity threshold.

An operational taxonomic unit or OTU is a grouping of entities (sequences in this context) that collectively satisfy a pre-defined similarity criterion. OTUs are simply pragmatic proxies for microbial “species.” Still the most widely used approach in marker-gene analysis, OTUs are commonly defined based on sequence similarity expressed as a percentage. A common threshold for defining OTUs is 97%, since similarity at this identity level is a widely accepted species definition for prokaryotes (Stackebrandt and Goebel 1994) and is sufficient to group real sequences and their close variants generated by sequencing errors into a single unit. However, the approach suffers from major drawbacks. In particular, 97% OTUs can merge sequences from organisms with very different roles in the microbiome. Conversely, ecologically cohesive groups may be split up into multiple 97% OTUs. Another key limitation is that *16S* gene sequences do not cluster perfectly into distinct OTUs. Ideally, an OTU, defined at a given similarity threshold, would contain sequences that all satisfy that threshold, and would contain no sequences that are also highly similar to sequences in a different OTU. However, this is not the case, and techniques are needed to untangle complicated relationships among sequences to yield coherent OTUs. A common approach is to define a *seed sequence*, either a best match to a reference database or a designated sequence within the sample, that is used to recruit other sequences into its cluster. In this context, all sequences with 97% or greater identity to a given seed sequence would be assigned to the seed’s cluster, even if they had similarity to other sequences not assigned to this cluster. However, the resultant grouping of sequences into OTUs can depend heavily on centroid selection.

Amplicon sequence variant (ASV) approaches present alternatives to the OTU approach that are more sensitive to patterns of sequence variation specific to a given dataset. For example, Swarm clustering (Mahé et al. 2014) is an alternative to fixed threshold OTU clustering. The Swarm application performs greedy single-linkage clustering by identifying seed sequences at random, then growing the corresponding clusters by adding all other sequences that satisfy a pre-specified identity threshold. Since this process is repeatedly performed until all clusters have stabilized, there is no dependence on the choice of initial seed. The greedy approach can occasionally generate clusters that are too large, so a second step is implemented which can break up these clusters. The refinement step considers the relative abundance of observed sequences within a cluster, and breaks connections in cases where abundant sequences (“peaks”) are connected only via less abundant intermediates (“valleys”).

Oligotyping (Murat Eren et al. 2013) is another type of clustering approach. It attempts to distinguish true sequence variation from sequencing errors by distinguishing mutations at different sites in the *16S* gene. Intuitively, if a site in the *16S* gene shows large amounts of variation within an OTU, then sequence differences at that site are likely to reflect true mutations. By contrast, sequencing errors are distributed randomly across the gene and will thus appear at sites that are otherwise invariable. Oligotyping implements this idea by calculating the Shannon diversity across all sites in the *16S* gene, and using only those sites with high information content (i.e. greater variation) to define types that can be considered as independent entities.

DADA2 (Callahan et al. 2016) is a recent clustering method that uses error-rate information to model the probability that an observed sequence variant i is produced by a sequencing error. The algorithm uses error probabilities and a Poisson model to determine if the abundance of the variant i is easy to explain as an error variant or a true sample sequence variant j . A low abundance of i relative to j yields a large p value, indicating that i is likely an error variant, while a high abundance relative to j suggests that i is a true sequence variant. The algorithm works iteratively from a single, large partition of all sequences, gradually subdividing the set until all subsets represent a single true sequence and its error variants.

The choice of clustering approach can have a dramatic impact on the number of units recovered and the corresponding downstream analysis. For example, a study by Koskinen et al. (2017) examined the number of groups recovered using either OTU clustering or the DADA2 pipeline. Using a PCR primer pair designed specifically to target archaea, the authors recovered between 320 and 490 OTUs (with a 97% identity cut-off) from a single stool sample, depending on the processing pipeline used, versus only eight DADA2 clusters from the same sample. Given the explicit models used by DADA2, the authors of the study viewed its predictions as a more accurate assessment of archaeal diversity in the microbiome.

Taxonomic Assignment

Since the principal value of marker-gene analysis is to assess the taxonomic profile and diversity of a given sample, assignment techniques are needed to make this link (Box 16.2). There are two basic strategies for taxonomic assignment: *supervised* approaches use a reference database to make assignments to specific taxonomic groups, whereas *unsupervised* approaches build representations from within the dataset (for example, OTUs with no assigned taxonomy), making no recourse to reference information. Hybrid approaches, where reference database assignments are complemented by unsupervised assignments of sequences with no high-confidence database match, can also be used.

Box 16.2 Diversity at Different Taxonomic Ranks

Bacteria and archaea are classified according to the traditional Linnean hierarchy of kingdom (or superkingdom or domain), phylum, class, order, family, genus, and species. However, microorganisms that share the same taxonomic name can have very different properties that impact their ecological role in the microbiome. For example, the genus *Escherichia*, which largely encompasses the named species *E. coli*, includes pathogens that affect a wide variety of hosts, including toxigenic foodborne pathogens such as strains O157:H7 and O104:H4. However, the group also includes strain K-12, which is harmless and has served as the model organism for much of molecular biology, and strain Nissle, which is recognized as having probiotic properties (Sonnenborn and Schulze 2009). Generalizations based on taxonomic names must be treated with great caution. A higher abundance ratio of the dominant phyla Bacteroidetes to Firmicutes in the gut has been associated with reduced obesity and other conditions (Ley et al. 2005), but both phyla encompass symbionts (for example, many short-chain fatty acid-producing members of the Lachnospiraceae family within Firmicutes) and pathogens (including *Bacteroides fragilis* within Bacteroidetes). Prokaryotic taxonomy is also highly contentious and fluid, in part because of the avalanche of new organisms that are being discovered and described both in the environment and in the laboratory.

Four reference databases are commonly used for taxonomic assignment: the RDP (Cole et al. 2014), Greengenes (DeSantis et al. 2006), SILVA (Quast et al. 2013), and the National Center for Biotechnology Information (NCBI) non-redundant (nr) sequence database (NCBI Resource Coordinators 2018). These databases differ substantially in size, coverage of different taxonomic groups, and the source of their taxonomic assignments. Indeed, other work (Balvočiūtė and Huson 2017) showed discrepancies in the taxonomic assignments provided by each of these resources.

Given the richness of reference databases, a common strategy is to perform taxonomic assignment of amplified sequences by comparing against a database of sequences with known taxonomic assignments. These assignments are typically done based on sequence similarity, which can be expressed in different ways. Similarity can be based on a simple search, where taxonomy is assigned to a sampled sequence based on its best match in a reference database. However, this approach suffers from limitations, particularly when a gene matches

S_1	AGCCTCGTTAACG
S_2	AGCTTCAAAGACG

	ϕ_1	ϕ_2
AA	1	2
AC	1	1
AG	1	2
AT	0	0
CA	0	1
CC	1	0
CG	2	1
CT	1	1
GA	0	1
GC	1	1
GG	0	0
GT	1	0
TA	1	0
TC	1	1
TG	0	0
TT	1	1

Similarity: vector dot product

$$Sim(S_1, S_2) = \sum_i \phi_1 \phi_2 = 11$$

Dissimilarity: Euclidean distance

$$D(S_1, S_2) = \sqrt{\sum_i (\phi_1 - \phi_2)^2} = 2.828$$

Figure 16.4 k -mer decomposition of a nucleotide sequence with $k = 2$. Two sequences, S_1 and S_2 , are decomposed into overlapping words of length 2. The corresponding $2^4 = 16$ word counts are expressed as a vector. Calculating similarity and dissimilarity values can then be computed from these vectors.

equally well to sequences from disparate taxonomic groups. A more refined approach is to use *phylogenetic placement*, which assigns taxonomy based on the optimal placement of a sequence in a phylogenetic tree of reference sequences. This approach can provide more precise taxonomic assignments, especially in the case of relatively novel sequences, but can be extremely time-consuming as likelihood scores may need to be calculated for all potential placements within a tree. Pplacer (Matsen et al. 2010) is an example of a popular phylogenetic placement software package.

k -mer decomposition is an alternative approach to taxonomic assignment that sacrifices some information within DNA sequences to dramatically increase the speed of an analysis. Instead of representing a DNA sequence as a complete series of bases, decomposition converts a sequence to counts of short, overlapping sequences of length k . For example, with $k = 2$ the sequence AGCCTCGTTAACG would be represented as $\{AA\} = 1$, $\{AC\} = 1$, $\{AG\} = 1$, $\{AT\} = 0$, and so on (Figure 16.4). Sequences can then be compared based on their k -mer distributions by expressing the distance between a pair of sequences in terms of the Euclidean distance between their frequency vectors. For taxonomic assignment, both the reference sequence database and the set of sampled sequences are converted into k -mer distributions, and statistical or machine learning approaches are used to compare these decompositions. The Ribosomal Database Classifier (Wang et al. 2007) is a widely used approach that uses a naive Bayes (NB) classifier trained on the RDP database to assign taxonomic information to sequences. Since the NB approach produces likelihood and posterior probability scores, sequence classifications can be constrained such that only predictions with an associated probability greater than a pre-defined threshold are accepted. Even if precise classification is not possible, assignment at taxonomic ranks such as class or phylum may have 100% associated confidence.

Calculating and Comparing Diversity

Many concepts initially developed in plant, animal, and fungal ecology have been adapted to microbial ecology. A particularly important aspect of ecological analysis is the assessment of biodiversity in a descriptive or relative context. The diversity of organisms at a given site can be a simple yet powerful indicator of the state of that site. For instance, a human-derived sample with few species or OTUs might suggest a state of poor health (e.g. dysbiosis); similarly, unexpectedly low diversity in a soil sample may reflect contamination or other environmental challenges. Single-site or alpha diversity (see Glossary) aims to express these concepts quantitatively. Another important question relates to the similarity in taxonomic diversity between two or more sites. Although alpha-diversity measures can be

contrasted between the sites, it is often of greater interest to identify which specific taxonomic groups are common between two or more sites, and which are distinct. In this context, the similarity between two sites may reflect common environmental factors, consistent interspecies interactions, or other factors. Beta-diversity measures are used to perform these comparisons.

Samples can be assessed and compared based on specific taxa of interest: for example, those taxa that disappear after antibiotic treatment. Since 16S analysis generates information about relative abundance, quantitative calculations of diversity can also be performed. Richness measures are based on taxonomic presence and absence, whereas diversity measures consider both the presence and the relative abundance (also known as the *evenness*) of taxa. Another distinguishing feature of diversity calculations is whether they are non-phylogenetic, in which all units (e.g. species or OTUs) are treated as equally dissimilar; or phylogenetic, in which the relative similarity of units is accounted for in the diversity calculation. In some analyses, it may be desirable to consider degrees of similarity such that closely related OTUs contribute less to overall diversity than do distantly related OTUs.

It is usually necessary to *normalize* samples in a study to compare them on an equal footing. Since each sample will have a different number of associated sequences, richness, and diversity calculations will be biased in favor of samples with either high or low read counts. *Rarefaction* in the context of OTU normalization involves randomly removing sequences from each sample until they each have the same sequence count, typically the count of the smallest sample in the dataset. However, rarefaction discards valid data. Another approach is to divide the count of each sequence in a sample by the total read count of the sample. While this approach preserves information, it introduces the problem of compositionality, in which observations are not independent owing to the normalization. An alternative approach is to use the zero-inflated Gaussian model implemented in the R package *metagenomeSeq* (Paulson et al. 2013). This method computes distributions for all OTUs with a strong tendency toward zero values because of the typical sparsity of OTU tables (i.e. most OTUs are absent from most samples). Each approach performs well under different circumstances, as recently reviewed by Weiss et al. (2017).

Alpha-diversity approaches consider the distribution of taxa in a single sample or at a single site. Rarefaction curves are used to assess the coverage of the total underlying diversity in a given sample. The key principle in most rarefaction strategies is to assess how much “new” diversity is added as the sample size is increased, typically by plotting a curve of taxon count versus sample size. If the count of taxa (e.g. OTUs) increases only very slowly as sample size increases, then the underlying diversity is likely to be well covered. Conversely, if observed taxa continue to increase with increasing sample size, then there is very likely poor coverage of the taxa that are present in the sampled habitat (Figure 16.5).

Richness measures consider the number of taxa present in a sample, without considering their relative abundance. The simplest measure of richness is to count the number of taxa observed in a sample. The phylogenetic variant of species counting is to sum all the branches in a tree that relate the different taxa to one another (Figure 16.6).

Richness measures are straightforward to use, but do not distinguish rare from abundant taxa, and therefore can be unduly influenced by the large number of rare OTUs that are found in most microbiome samples. Diversity measures consider both the count of species and their relative abundance. For example, the Shannon diversity index quantifies the entropy of a sample using the following formula:

$$H' = - \sum_{i=1}^R p_i \ln p_i$$

where i iterates over each species with indices 1 ... R , and p_i is the proportion of species i relative to all sampled species. The maximum value of the Shannon index increases with increasing R , and with increasing evenness of the sample (i.e. when $p_i = 1/R$ for all i). The Simpson index,

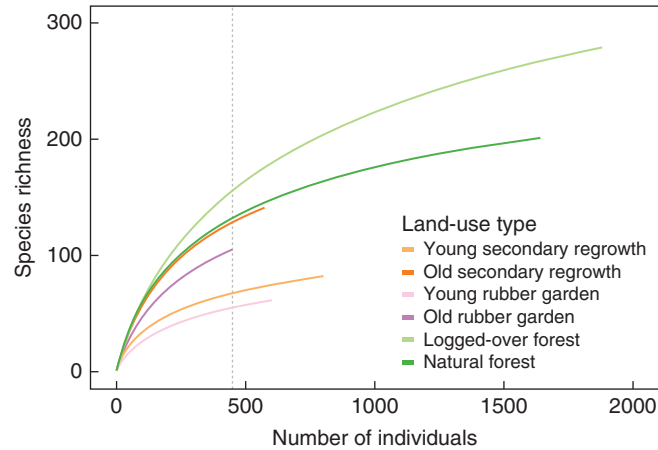


Figure 16.5 Rarefaction curves for microbial communities sampled from six different habitat types. Two samples (young secondary regrowth and young rubber garden) are approaching asymptotes more rapidly than the other samples, suggesting these habitat types have been sampled more completely than the other four. Source: Labrière et al. (2015; doi.org/10.1371/journal.pone.0140423.s001).

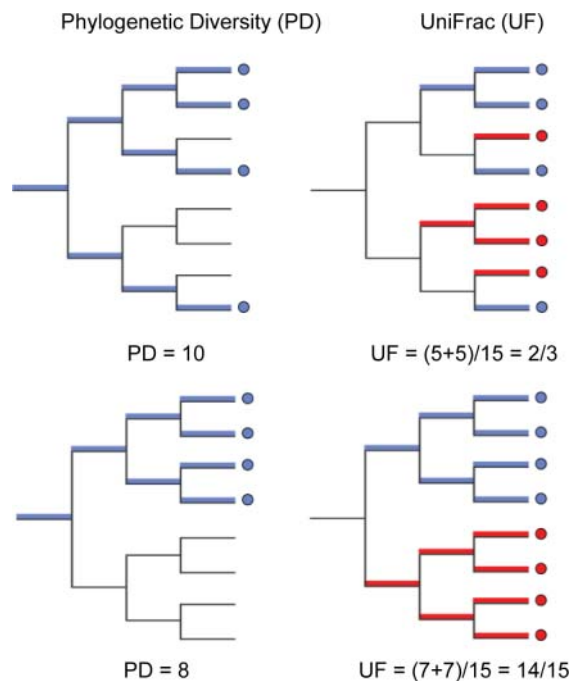


Figure 16.6 Unweighted phylogenetic alpha- and beta-diversity measures. Left: the phylogenetic diversity of a sample is equal to the sum of branch lengths from all leaves of the tree that are represented in a sample, up to the root of the tree. The example on top covers a wider phylogenetic range than the bottom example, and therefore has a larger diversity score. Right: the UniFrac score considers the number of branches unique to one sample or the other (indicated with red and blue), divided by the total number of branches in the tree. In the top example, several ancestral branches are shared, leading to a relatively low UniFrac score. In the bottom example, only the root is shared between the two, which yields a much higher beta-diversity score. Source: Karlsson et al. (2015; doi.org/10.1371/journal.ppat.1005225.g004).

which expresses the probability that two taxa drawn at random from a sample will be the same, is also commonly used:

$$\lambda = \sum_{i=1}^R p_i^2$$

Weighted phylogenetic diversity measures, which adjust the contributions of different tree branches according to the abundance of taxa they cover, complement the phylogenetic richness measures described above.

Beta-diversity approaches compare taxonomic distributions between samples or sites to yield an expression of dissimilarity. Beta diversity is typically calculated between all pairs of sites, with multivariate methods then used to summarize the pairwise patterns. These measures can be classified in a manner similar to alpha-diversity measures: qualitative measures are akin to richness indices, while quantitative measures correspond to diversity measures. Phylogenetic

and non-phylogenetic variants exist as well. Qualitative measures consider only the presence and absence of shared taxa between two samples. The simplest such measure is the Jaccard index (J), which for a pair of sites A and B is:

$$J_{A,B} = A \cap B / A \cup B$$

where $A \cap B$ is the intersection (i.e. the set of taxa that are shared between A and B) and $A \cup B$ the union (i.e. the set of taxa that are found in at least one of A and B). These quantities are equal if both sites have identical sets of taxa, in which case $J_{A,B} = 1$. The Jaccard index is the direct counterpart to species richness, and $1 - J$ is a widely used measure of dissimilarity. A widely used qualitative measure of diversity is unweighted UniFrac, first introduced by Lozupone and Knight (2005). This method maps taxa onto a phylogenetic tree, and divides the lengths of all branches covered by both samples by the total length of all branches in the tree. The motivation behind UniFrac and the plethora of other phylogenetic diversity-based approaches (Parks and Beiko 2013) is to assign lesser importance to differences that are due to closely related organisms, assigning greater importance to differences at higher levels of divergence. Quantitative measures consider the relative abundances of taxa when making a comparison. The Bray–Curtis dissimilarity index (BCI) is a quantitative variant of the Jaccard index:

$$BCI_{A,B} = 1 - \frac{2C_{A,B}}{S_A + S_B}$$

where $C_{A,B}$ is the shared count of taxa, i.e. if a given taxon T is present five times in sample A and seven times in sample B , then $C_{A,B}$ will be incremented by 5. S_A and S_B are the total counts of all taxa in both samples. If two samples have the exact same counts for all taxa, then the BCI will be equal to 0. Weighted UniFrac considers shared branches as with its unweighted counterpart, but weights each branch according to the relative abundance of taxa covered by that branch.

Calculating beta-diversity measures between all pairs in a set of samples yields a matrix that expresses the similarity or dissimilarity of each pair of samples, and can be summarized in several different ways. A common approach is to use methods that extract shared patterns of covariance across many samples, allowing low-dimensional visualization of the high-dimensional matrix. Widely used examples include principal coordinate analysis (PCoA), also known as multidimensional scaling, and non-metric multidimensional scaling (NMDS). PCoA is an ordination technique that identifies linear correlations between the calculated dissimilarity values in the matrix. Orthogonal linear correlations are termed principal coordinates, with the importance of each coordinate reflecting the amount of variance it captures from the original dataset. Eigenvalues indicate how much of the original variance in the dataset is reflected in a given principal coordinate. Principal coordinates are ranked in decreasing order of importance, and can then be plotted against one another. Sets of samples that cluster together in the resulting plot are likely to share common patterns of diversity, then may be interpreted in light of an environmental or other variable. PCoA is closely tied to the more familiar principal component analysis (PCA), but whereas the starting point for PCA is a data table, PCoA uses a matrix that can be based on any beta-diversity measure. NMDS generates axes without the linearity assumptions of PCoA and operates by embedding samples in a two- or three-dimensional space to minimize a “stress” criterion. Stress is expressed as a mismatch between the rank order of dissimilarity values in the matrix and the corresponding ranks of distances in the NMDS plot. The NMDS approach iteratively adjusts the positions of points in the plot to minimize stress (Figure 16.7).

The plethora of diversity formulae available highlights the absence of obvious and objective criteria to judge and compare sample diversity. There is no single best measure for all situations. For example, Parks and Beiko (2013) simulated different types of shift in community structure, and demonstrated that different measures of phylogenetic beta diversity were most effective under different simulated conditions. Since the detection of patterns of interest can depend on the choice of method used, it is worthwhile to consider several methods that tend to give contrasting results.

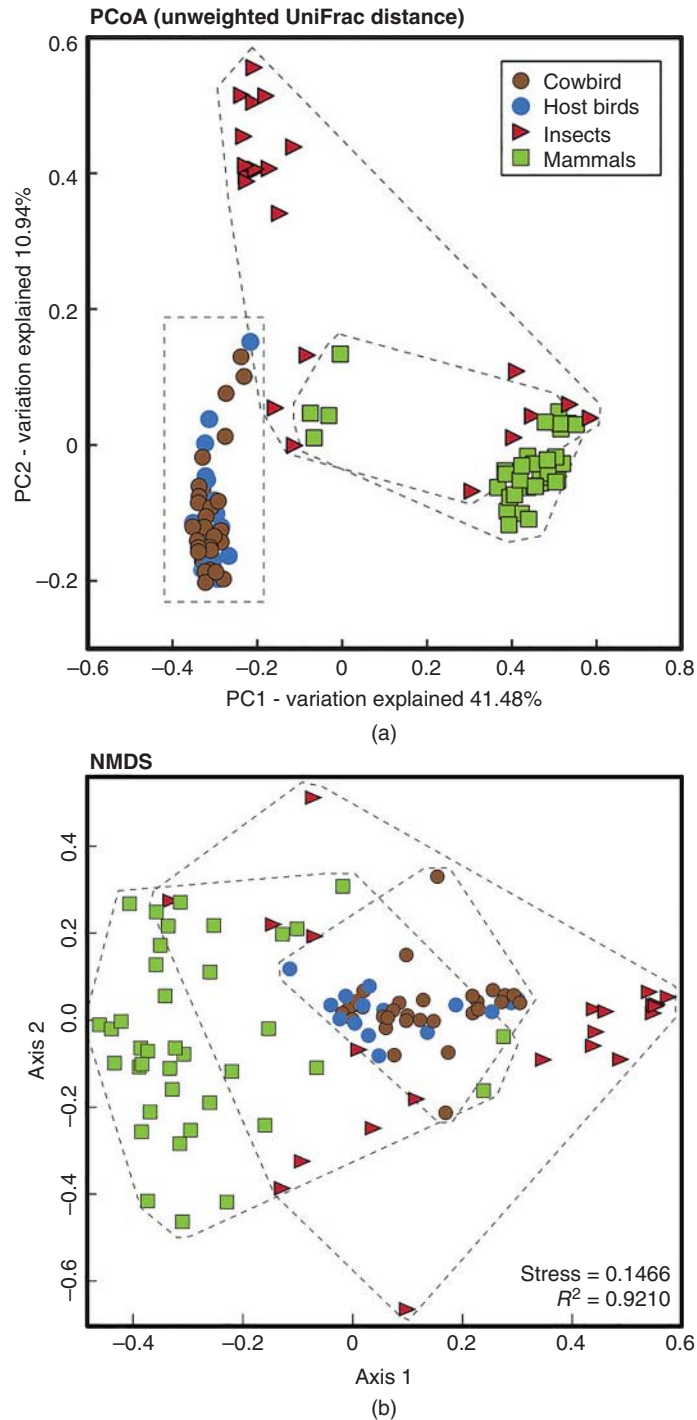


Figure 16.7 Principal coordinate analysis (a) vs. non-metric multidimensional scaling (b), showing two contrasting views of the gut microbiota of brood-parasitic brown-headed cowbird (*Molothrus ater*) and three comparator groups. Dashed lines are bounding boxes for each sampled group. Source: Hird et al. (2014; doi.org/10.7717/peerj.321/fig-5).

Associations with Metadata

Many statistical and machine learning approaches have been used to associate taxonomic information with environmental metadata parameters such as originating body site, soil pH, and nutrient concentrations. A variety of two-sample and multiple sample tests such as the *t*-test and analysis of variance (ANOVA) (and permutational variants thereof, such as PERMANOVA) have been used to identify significant distinctions between and among samples. The large number of sequences and OTUs recovered from marker-gene analyses raises a multiple test problem in statistical analysis: with so many potential predictors, some are likely to produce significant relationships due to chance. In such cases, it is essential that a multiple test correction, such as the Bonferroni or false discovery rate correction (see Chapter 18), be applied. Effect size must be considered in addition to statistical significance, as statistical results may have very small *p* values but uninteresting differences between different types of sample. STAMP (Parks et al. 2014) and LEfSe (Segata et al. 2011) are two programs that aim to address several of the statistical challenges in metagenomics. LEfSe works by assessing differences between sample types using linear discriminant analysis, then identifying those features that contribute the most to the distinctions between classes. LEfSe produces visualizations that highlight those taxa that are over- or under-represented in different classes of samples (Figure 16.8).

Machine learning approaches have also been used to find associations between the microbiome and environmental data. The high dimensionality of *16S* sequence and OTU data requires methods that can process many inputs, and potentially the use of up-front feature selection methods to pare down the candidate groups of interest. Knights et al. (2011) applied several approaches that span statistics and machine learning, including random forests and elastic net classifiers, to the classification of several reference metagenomic datasets using OTUs as input. No clear winner was found, consistent with the “no free lunch” theorem that loosely states that no single classifier will be globally most effective for all types of classification problem. Ning and Beiko (2015) applied feature selection and random forest classifiers to the classification of oral microbiome samples, and found that flexible phylogenetic definitions of groups were often more effective than strictly defined OTU thresholds, highlighting the limitations of OTUs in microbial community analysis.

Marker-gene-based approaches exploit genomic diversity to build comprehensive views of microbial diversity. However, the limitations described in this section need to be considered when designing a marker-gene survey and interpreting the results. Comparisons of different approaches for taxonomic profiling of microbial communities have raised significant concerns about the accuracy of PCR-based analyses (Schirmer et al. 2015). Beyond these technical considerations, another significant pitfall lies in the assignment of putative ecological roles to taxa or OTUs based on their taxonomic name alone. Given the capacity of microorganisms for rapid ecological diversification, a named species or genus may comprise many distinct lineages that perform different tasks, and may sometimes exhibit strong negative correlations in their abundance. In spite of these limitations, marker genes such as *16S* genes are useful tools for generating hypotheses about important drivers of change in community diversity and function.

Metagenomic Data Analysis

The utility of marker genes lies in their ubiquitous distribution across all sampled taxa; however, their abundance or taxon assignments provide little direct evidence of community function. The inferred presence of specific taxonomic groups may imply the presence of specific biochemical functions, but direct functional evidence needs to be obtained through different means. The first metagenomic (as opposed to marker-gene based) analysis (Handelsman et al. 1998) used cloning and expression in vector libraries rather than shotgun sequencing to assess the range of functions present in a microbial community. However,

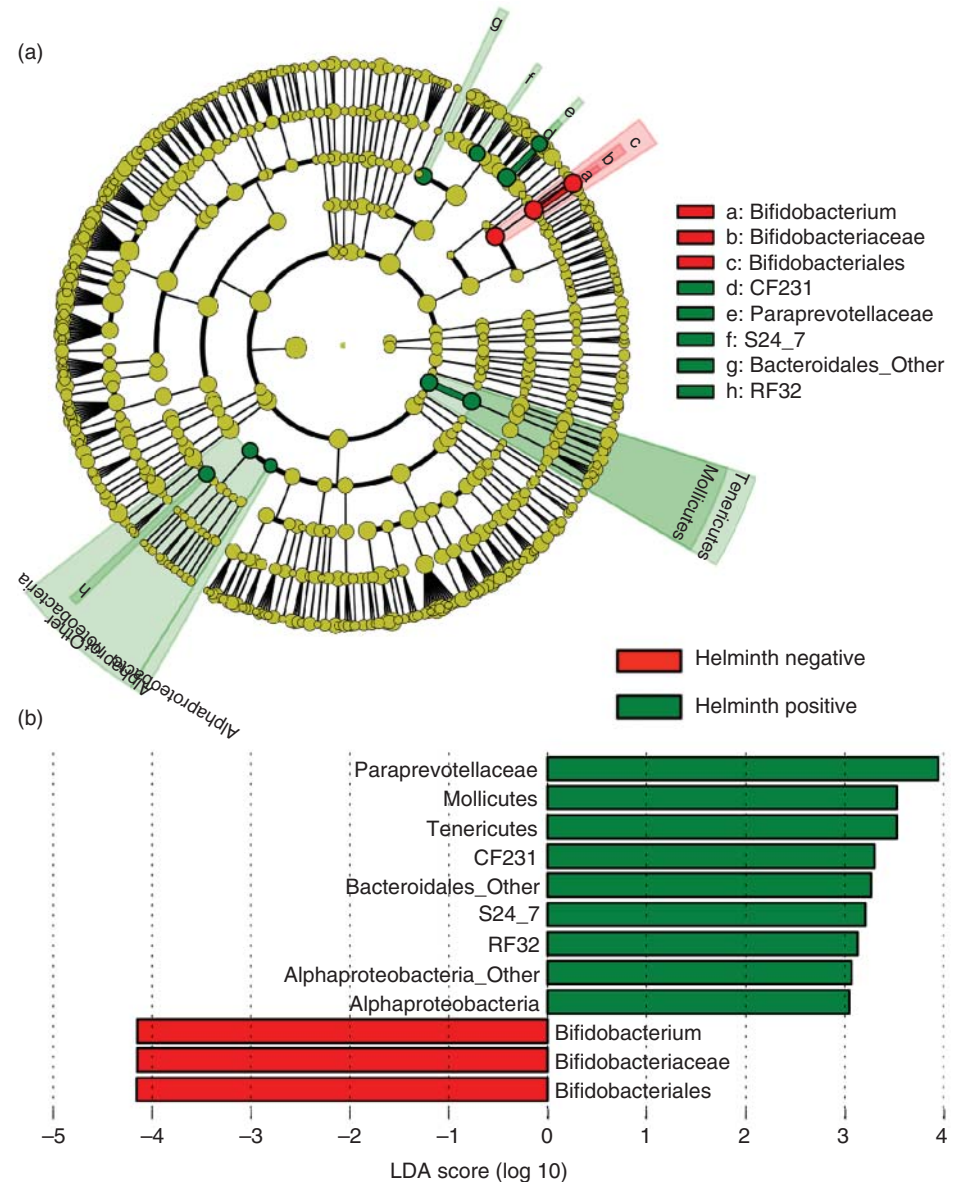


Figure 16.8 Visualizing the differences between two groups of gut microbiome samples in LefSe. In this example, the phylogenetic distribution of operational taxonomic units from the guts of individuals with (green) or without (red) helminth colonization (a). Red dots indicate phylogenetic groups and specific taxa that are over-represented in the helminth-negative group, while green groupings indicate the reverse. (b) Magnitude of effect size, according to linear discriminant analysis (LDA) score. Source: Lee et al. (2014; doi.org/10.1371/journal.pntd.0002880.g003).

cloning approaches are expensive, and high-throughput sequencing (which the remainder of this section will address) rapidly became the standard approach to characterize the functional and taxonomic diversity of a microbial community. The level of detail afforded by DNA sequencing allows a more complete enumeration of candidate functions within a microbial community, which can then be used to build models of community function.

Predicting Functional Information from Marker-Gene Data

Recently developed methods such as Phylogenetic Investigation of Communities by Reconstruction of Unobserved States or PICRUSt (Langille et al. 2013) aim to predict the

metagenomic functional repertoire from a marker-gene survey. These predictive approaches exploit the growing number of sequenced microbial genomes, which provide linkages between markers and candidate functions. PICRUSt is based on the insertion of sampled *16S* sequences into a reference tree that includes *16S* sequences from all microbial genomes. The functional predictions for this environmental sequence will be based on the gene content of genomes that are nearby in the *16S* tree. The entire set of functions in the metagenome is predicted based on the sum of all functions associated with each environmental sequence, weighted by their relative abundance in the sample.

The accuracy of PICRUSt is dependent on two related factors: the availability of suitable genomes in the reference database and the conservation of functional traits. Accuracy is highest when there are many sequenced genomes that are closely related to constituents of the metagenome. Initial validation across several habitats showed the highest accuracy in predicting gut metagenomes; this is due in large part to the intensive sequencing of many reference genomes from the human gut microbiome. Conversely, PICRUSt accuracy in habitats with far fewer reference genomes available, such as hypersaline environments, was much lower. The second key driver of accuracy in PICRUSt is the degree to which different functional traits are conserved. Core functions such as ribosomal proteins are conserved across much greater taxonomic distances, and are therefore much easier to predict. Lower average accuracy is obtained for traits that are less well conserved, such as habitat-specific transporters, since these traits will be much more unevenly distributed across a small taxonomic range. Uncertainty in predictions is expressed as a confidence interval around the mean prediction, with smaller confidence intervals corresponding to more confident predictions. Although PICRUSt can only generate predictions of the functional breakdown of a metagenome, these predictions can be used in many of the tools that are typically applied to real metagenomic data.

Metagenomic Analysis Protocol

As in the case of *16S* analysis, many options are available to perform each relevant step of metagenomic analysis. Since metagenomic datasets comprise reads sequenced at random from many genomes, some aspects of metagenomic analysis resemble those used in genome assembly and analysis. However, the heterogeneous origin of sequence reads confounds steps such as sequence assembly, and creates a need for taxonomic assignment. Microbiome Helper (Comeau et al. 2017) is a software package for metagenomic analysis that integrates a suite of standard tools into an overall workflow. Other metagenome workflow tools include the metagenomic version of “A Modular, Open-Source whole genome assembler” (metAMOS; Treangen et al. 2013) and MEtaGenome ANalyzer (MEGAN; Huson et al. 2016). Although the remainder of this section is presented as a plausible linear workflow for metagenomic data analysis, different aspects such as gene annotation, sequence searching, and functional assignment can be accomplished in the same step. For example, if metagenomic genes are annotated via sequence alignment against a reference database, then functional annotations of the reference genes can be assigned to predicted genes at the same time.

Quality Control and Merging of Paired-End Reads

Since metagenomics involves sequencing of random DNA fragments, sequences from phage, virus, and eukaryotic host organisms may be removed from a dataset if they are not of interest in the study. These are typically removed through comparisons against a reference database, using an efficient read mapper such as Bowtie2 (Langmead and Salzberg 2012). Metagenomic reads that match sequences from the reference database are removed from the metagenomic dataset.

Since some DNA sequencing platforms sequence both ends of a DNA fragment, the two corresponding “paired-end” sequences can be stitched together in order to retain information about their proximity to one another in the source genome. Depending on the fragment size

and read length, these reads may even overlap, in which case contiguous sequences can be obtained using read-stitching programs such as the Paired-End reAd mergeR (PEAR; Zhang et al. 2014). If reads do not overlap, then there will be unknown sequence between the two paired ends, but their spacing relative to one another will still be known.

Assembly

Although metagenome analysis can be performed directly on DNA sequence reads, longer contigs provide a great deal more information about linkage between genes, and provide better statistical information for inference of taxonomic and functional distributions. One key assumption of standard genome assemblers is that all reads originate from the same clonal organism. With metagenomic data this is clearly not the case, and assembly is confounded by the need to identify subsets of reads which *should* be assembled. If closely related strains are present in a given sample, it may be impossible to identify the originating strain of a particular read, and to perform cross-strain assemblies where highly conserved regions are interspersed with strain-specific regions of high variation or differential genome content. Key indicators of assembly quality such as N50 (a measure of contig length distribution) are irrelevant in the assessment of assembly quality, since rare organisms in the sample will not be well covered by sequencing reads, and will therefore tend to assemble into short contigs or remain as unassembled fragments. Long-read sequencing offers a tremendous advantage in these cases, as reads that are several thousand nucleotides in length offer better information for assembly purposes, and can span many difficult to assemble regions, including those that contain repeat sequences, and strain-specific gene content. Although paired-end Illumina reads are shorter, they can nonetheless still produce larger contigs than unpaired reads alone.

Metagenome assembly can be simplified by binning sequence reads and contigs into multiple subsets which can then be assembled further. This binning can be accomplished by comparing contig properties – nucleotide composition and relative abundance, in particular. Since fragments of the same genome often have similar patterns of nucleotide composition (i.e. *k*-mer distributions), contigs with similar distributions can be assumed to derive from the same originating genome. Abundance-based binning approaches are based on the idea that organisms with similar abundance in a sample should generate metagenomic contigs with the same relative abundance. These approaches have allowed the reconstruction of complete or near-complete genomes from the environment. For example, Albertsen et al. (2013) used differential abundance profiles and tetranucleotide frequencies to assign assembled scaffolds to 31 distinct population bins from a wastewater reactor sample, then assembled the reads within the 13 most abundant bins into draft genomes. Through this method, the authors were able to distinguish and reconstruct four high-quality genomes from the little characterized TM7 phylum. However, binning approaches are limited by the amount of information they can extract from short reads, and customized metagenome variants of existing genome assembly tools such as Ray Meta (Boisvert et al. 2012) and metaSPAdes (Nurk et al. 2017) have emerged.

Gene Annotation and Homology Searching

Searching for homologous sequences against reference sequence databases is an essential step in the functional and taxonomic annotation of metagenomes. In general, only regions that encode proteins or structural RNAs (such as rRNAs and transfer RNAs) are targeted for homology-based annotation. An obvious challenge in gene annotation for metagenomic datasets is that metagenomic reads and contigs are highly likely to contain fragments of open reading frames (ORFs) and may be missing 5' regions, 3' regions, or both. In some cases, there may be insufficient information in a given DNA sequence read to make any annotation call, but, even if a gene fragment of substantial length is present, the correct start or stop codon may be missing. Although not unique to metagenomic datasets, the challenge of identifying novel genes with no known homologs in existing sequence databases is especially acute in

metagenomic gene annotation. Complementary approaches, such as identifying ORFs based on codon usage patterns, can therefore be important as well. For example, Zhu et al. (2010) extended the widely used GeneMark gene annotation software package (see Chapter 5) to develop MetaGeneMark, which applies compositional statistics from reference microbial genomes to metagenomic reads.

Homology-based gene annotation can be achieved in several ways. Read-mapping approaches based on the Burrows–Wheeler algorithm for sequence alignment can be applied to perform tiling of metagenomic reads and contigs against reference genomes. In many cases, especially where novel taxonomic groups are present in a metagenomic sample, the similarity between the query and reference sequences may be too low to be recognizable using Burrows–Wheeler approaches such as the Burrows–Wheeler aligner (BWA) (Li and Durbin 2009) and Bowtie 2 (Langmead and Salzberg 2012). More dissimilar sequences can be recognized using the BLAST suite of algorithms (Altschul et al. 1997) – specifically, BLASTN for direct nucleotide–nucleotide comparisons and BLASTX for comparisons of a reference database of proteins against a six-frame conceptual translation of metagenomic sequences (see Chapter 3). An advantage of BLASTN is that it can recognize intergenic sequences as well as protein-coding genes, and thus has the potential to identify genomically close sets of genes (often referred to as linked or syntenic genes) instead of individual genes with no positional context. However, BLASTX searches are conducted in the more highly conserved protein sequence space, which makes BLASTX more suitable for detecting remote homologs. This sensitivity comes at a significant cost, as the six-frame translation and dynamic programming elements of BLASTX can be computationally time-consuming for the comparison of large datasets against large reference databases. Recent methods to accelerate sequence alignment have yielded speed-ups of up to four orders of magnitude relative to BLASTX. In one such example, DIAMOND (Buchfink et al. 2015) was used successfully to compare a set of arctic permafrost environmental samples against the NCBI nr database in less than 3 hours on a single workstation, as compared with 800 000 CPU hours for BLASTX. DIAMOND achieves this speed-up using a range of optimizations, including a different strategy to seed alignments and a reduced amino acid alphabet (11 amino acids instead of the usual 20).

For highly sensitive searches, approaches such as hidden Markov models (HMMs) can be used (see Box 5.3). HMM residue frequencies as well as site-specific insertion and deletion probabilities in proteins are especially popular for the annotation of short protein functional sequence motifs owing to their increased sensitivity. HMMs are typically trained from reference sequence databases, with HMMs potentially corresponding to protein domains or other functional groupings. Functional Ontology Assignments for Metagenomes (FOAM) (Prestat et al. 2014) is a set of over 70 000 HMMs trained from Kyoto Encyclopedia of Genes and Genomes (KEGG) orthology groups, which can serve as a reference database for metagenome sequence comparisons.

Taxonomic Assignment and Profiling

The homology-based procedures described above yield database matches with corresponding functional labels; if the match between query and reference sequences is high enough, then it may be justifiable to assign the corresponding taxonomic label (e.g. genus and species) to the metagenomic sequence. Performed over an entire metagenomic dataset, this approach could be used to build a global taxonomic summary of the sample, and to precisely assign taxonomic information to specific reads and contigs (and their corresponding functional information; see Functional Predictions). However, metagenomic datasets present several obstacles to this general approach. First, as in other analyses, novel taxa that lack representation in reference databases will not be classifiable at the lowest taxonomic ranks: instead, higher taxonomic ranks such as order, class, or phylum may represent the most precise classifications that can be made. Second, as different types of genes show different degrees of conservation, it may

be possible to distinguish some genes (e.g. rapidly evolving metabolic genes) at the species or strain level, but not highly conserved genes such as those that encode ribosomal proteins or 16S rRNA. Third, as prokaryotic genome sizes can vary by over an order of magnitude, larger genomes will be over-represented in a sample relative to smaller ones, skewing the predicted taxonomic distribution. This limitation mirrors the copy number problem in 16S gene analysis. Finally, mobile genetic elements such as plasmids and genomic islands can readily be moved between distantly related organisms through the process of lateral gene transfer, and a gene that was recently acquired into a recipient may still be classified to the donor organism. Therefore, some classes of genes must be treated with caution.

k-mer decompositions have been exploited by many software packages to achieve rapid taxonomic classification of metagenomic sequences. Depending on the method used to compare distributions, *k*-mer-based approaches can be many orders of magnitude faster than sequence alignment approaches. Distance calculations, interpolations of higher order *k*-mer abundances and machine learning approaches such as NB (Rosen et al. 2011) have all been used to compare the composition of metagenomic sequences against reference sequence databases. In general, these methods tend to be faster but less precise than BLAST algorithms for taxonomic assignment. Kraken (Wood and Salzberg 2014) uses an alternative approach to *k*-mer modeling and matching that yields high precision while retaining significant speed-ups relative to BLAST. The key to Kraken is the decomposition of sequences into long (by default, 31 nt) *k*-mers: rather than trying to compute distances between profiles (which would be practically impossible with 4^{31} 31-mers), each identified *k*-mer is treated as a potential taxonomic marker in its own right. Precomputed phylogenetic trees are inferred from genomes in a reference database, then each observed *k*-mer is mapped to the last common ancestor of all organisms in which that *k*-mer is found. For example, if a given 31 nt sequence was observed only in the order Enterobacteriaceae, then it would be treated as distinct for that group. Metagenomic reads or contigs are then decomposed into *k*-mers, which are then compared against a reference tree. The sum total of evidence from the set of *k*-mers is then used to identify the most probable originating lineage of the corresponding metagenomic sequence. In trials with simulated metagenomic data, Kraken showed nearly equal sensitivity to the MegaBLAST algorithm, with a nearly 1000-fold speed-up.

Given that metagenomic samples contain fragments of the 16S gene, it is possible to extract these gene sequences and perform taxonomic classifications that can be compared against standard 16S surveys. A significant advantage of this approach is the elimination of amplification bias, since PCR primers are not used in metagenomic data generation. However, taxonomic profiling in this manner discards nearly all of the other metagenomic data, and comparisons across different fragments of the 16S gene can be difficult. Furthermore, one of the key drivers of 16S use in amplicon studies is its tractability for amplification, but metagenomic data provide information about dozens of core genes that are single copy and highly conserved. Genomic databases comprising tens of thousands of taxonomically referenced genome sequences allow detailed investigation of the taxonomic patterns associated with these core genes.

Subset-based approaches offer another route to performing taxonomic assignments. These methods produce taxonomic summaries of metagenomes based on the annotation of a small set of core genes that are present either in most genomes or in targeted subsets of genomes. Two widely adopted methods, PhyloSift and MetaPhlAn, illustrate the contrast between these approaches. PhyloSift (Darling et al. 2014) uses a set of 37 widely distributed protein families, plus the 16S gene, that show high phylogenetic congruence. These sequences are first identified in a metagenomic dataset via fast sequence alignment searches against the reference database. The reference set for each gene is maintained as a model multiple sequence alignment and a phylogenetic tree. Once matching sequences are identified, they are aligned precisely to the model alignment using HMMs, and inserted into the reference tree using Pplacer. The summary of phylogenetic placements leads to the inferred taxonomic composition of the metagenome. By contrast, MetaPhlAn (Segata et al. 2012) takes the

opposite approach by building a reference database of >400 000 genes that are “core” to genomes within a specific clade, and excluded from all genomes outside that clade. Given the small size of the reference set of genes to the set of all genes, homology-based processing of a metagenomic dataset can be achieved very quickly.

Of course, as a researcher wishes to learn not only “who is there” and “what they are doing,” but additionally “who is doing what,” then they must match each metagenomic read or contig to reference databases using some combination of compositional assessment and sequence alignment. Kraken achieves this through its unique compositional approach, and BLAST offers a highly sensitive but relatively slow sequence alignment approach. A limitation of BLAST is that the best match of a metagenomic sequence to a database may not represent the correct taxonomic classification, either because the sequence is novel or because there are many database matches with nearly equal scores. Mirroring the phylogenetic approaches adopted by Kraken and PhyloSift, some methods use phylogenetic mapping of *all* predicted protein sequences to make taxonomic predictions. MEGAN uses an effective last-common-ancestor mapping to estimate the taxonomic rank of a metagenomic sequence.

Functional Predictions

Although “function” can be a difficult term to define, in general its use in metagenomics refers to the capacity of organisms in the microbiome to perform enzymatic reactions, engage in ecological interactions with other organisms including the host, and construct important molecular structures such as the flagellum. Function can be categorized in many ways; for example, the top level of the Gene Ontology (see Chapter 13) hierarchy breaks functions into the three categories of biological process (a series of functions, for example a biochemical pathway), molecular function (the type of reaction catalyzed by a given enzyme), and cellular compartment (the location in which a protein’s action takes place, for example in the cytosol or the periplasmic space). Functions can also be described at different levels or organization: for instance, KEGG can define function at levels including function, pathway (a collection of functions), and module (a collection of pathways). The simplest way to summarize functions for a metagenome is to carry out a sequence alignment search against a reference database, and summarize the search results in terms of the presence or absence, relative abundance, or diversity of a particular function in the metagenome. However, this basic approach has several limitations.

The first limitation is the possibility of false-negative annotations of proteins. In many cases, homologous proteins that are highly similar can in fact have different functions: for example, transporters and efflux pumps with similar amino acid sequences can act on different substrates and with different reaction kinetics, making it very difficult to predict a molecular target for these proteins in the absence of a large and well-characterized reference sequence database. At the same time, many predicted proteins from a metagenome may match hypothetical proteins with no validated function or may match no reference proteins at all. These proteins will therefore contribute no direct information to the functional summary. The use of different reference sequence databases will lead to different balances between false-positive predictions and unannotated genes; the Swiss-Prot database (UniProt Consortium 2018), which comprises only sequences with manually annotated and reviewed functions, will produce relatively few functional annotations, but these will generally be of high quality. Conversely, the KEGG database (Kanehisa et al. 2017) has a higher coverage of functions, the majority of which have not been experimentally validated or manually reviewed. KEGG will consequently tend to produce more incorrect annotations. Specialized, curated sequence databases can be used to focus on particular functions, including CAZy (Cantarel et al. 2008) for carbohydrate-active enzymes and Comprehensive Antibiotic Resistance Database (Jia et al. 2017) for antimicrobial resistance genes.

Another problem with functional annotation is the promiscuous assignment of pathways based on the presence of relatively few steps in the pathway, or because of the presence of

a single function in multiple pathways. Naive prediction will produce an excessively large set of pathways, many of which are functionally irrelevant in the sample. MinPath (Ye and Doak 2009) was developed to address this limitation by identifying a minimal set of pathways that can cover all annotated functions in a metagenome. Applying MinPath to reference metagenomes reduced the number of predicted pathways by as much as 50%. HUMAnN (Abubucker et al. 2012) uses MinPath as part of a pipeline to predict, then filter, predicted functions and pathways, and report the presence and relative abundance of pathways.

Statistical Associations

As with marker-gene analysis, an important goal of metagenomics is to discover statistical associations between inferred patterns of biodiversity and environmental parameters. Taxonomic associations and co-occurrence patterns can be augmented with functional diversity and changes in the relative abundance of closely related organisms that have different ecological roles. Jonsson et al. (2016) recently reviewed a range of statistical tests and applications for comparing functional distributions across metagenome samples. This study found that the tools used in transcriptional profiling to assess differential expression between two or more samples, DESeq2 (Love et al. 2014) and edgeR (Robinson et al. 2010), were most effective relative to standard statistical procedures such as the *t*-test. The availability of functional data also offers the opportunity to perform comparative analyses of metabolic networks. BiomeNet (Shafiei et al. 2014) uses unsupervised Bayesian methods to identify subnetworks that differentiate types of metagenomic samples. Since the division of metabolic networks into pathways can be somewhat arbitrary, approaches that do not presuppose particular boundaries on pathways can be more sensitive to functional variation.

Metagenomics is a powerful technique that can generate a comprehensive profile of a microbial community sample. The diversity of sampled sequences can reveal critical strain-level diversity and variation, and the assembly and annotation methods described above have been used to reconstruct genomes from previously uncharacterized phyla. Another valuable application of metagenomic analysis is in the discovery of novel gene variants that have different substrates or activity levels. Metagenomic data inherit many of the limitations of genome analysis, including short-read assembly and the challenges of homology-based functional annotation. Accurate long-read sequencing will significantly reduce assembly problems. However, given the relatively high cost of long-read sequencing, sequencing effort that is wasted on host DNA incurs a substantial additional cost. Although highly effective at giving a functional cross-section of a microbiome sample, metagenomics provides no evidence about transcriptional responses to environmental stimuli or changes. As with marker-gene analysis, it can be difficult to identify which microorganisms inferred to be present in a microbial community sample are “true” members of the community, i.e. metabolically active and interacting with other organisms, versus those that are ephemeral in a given habitat. Making these distinctions requires targeted approaches that consider functions and ecological roles of the constituents of the microbiome. Hanage (2014) outline some of the data interpretation challenges involved in metagenomic data analysis.

Other Techniques to Characterize the Microbiome

So-called multi-omic datasets aim to address some of the limitations of marker-gene and metagenomic analysis. These include, for example, measurements of transcript and protein expression. Stable diversity, reflected by similar metagenomic profiles, may conceal significant transcriptional variation in response to medication, rapid temperature shifts, day/night patterns, and other environmental parameters. Other approaches, such as metabolomics, track the metabolic outputs of the microbiome rather than molecular sequence data (see Chapter 14). Finally, methods to tease apart the complexity of the metagenome can enable

precise characterization of subsets and individual constituents of the microbiome. Franzosa et al. (2015) outlines many of these multi-omic techniques in detail.

Pioneered in the early 2000s, metatranscriptomic datasets characterize global microbial gene expression using approaches such as RNA-seq. Metatranscriptomics makes use of differential expression analysis to identify expressed genes and differences among samples. DESeq2 (Love et al. 2014) is a program that is commonly used to identify transcripts with differential abundance between multiple samples, by mapping sequence reads to reference sequences then computing the fold change and statistical significance of differences. Metaproteomics bypasses nucleotide sequencing and uses protein digestion and mass spectrometry to identify fragments that map to reference genomic and metagenomic sequences. Proteomic techniques (see Chapter 11) can be discovery based, in which case information about the presence of all proteins is sought, or targeted, where a small subset of proteins is selectively monitored. The latter provide more precise information about proteins that may be critical in community function, or that may serve as candidate biomarkers.

One of the key roles of microorganisms in many settings is to produce metabolites that can serve as energy sources for other organisms. Through their enzymatic activities they can also transform extracellular compounds, changing their function and role. Meta-metabolomics identifies unique spectral patterns generated by different metabolites in a system, then matches these to known metabolites via comparisons against a reference database. Metabolite profiles are characterized by identifying spectral peaks and matching these peaks against reference spectral databases using software such as XCMS (Smith et al. 2006), as described in Chapter 14.

Stable isotope probing can be used to track the flux of metabolites between members of a community. Isotopes of atoms that are widely used in metabolism such as ^{13}C and ^{15}N can be used as tracers, as an organism supplied with these isotopes will integrate them into its biomolecules, then transfer them to other organisms. Berry et al. (2013) used stable isotope probing to track the flow of metabolites from a mouse host to intestinal microbiota, and fluorescent in situ hybridization (FISH), which can measure the expression of specific marker genes, to identify *Akkermansia muciniphila* and *Bacteroides acidifaciens* as important consumers of host proteins.

Given the diversity of many environments, it is inevitable that some microbial taxa will be poorly represented in metagenomic samples, yielding partial or no assemblies and incomplete characterizations. Subdividing or subsampling microbiome samples can increase the recovery of taxonomic groups of interest. Even if single isolates cannot be grown in pure culture, in many cases enrichment cultures with lower diversity can be grown. If an enrichment culture cannot be further purified, then the corresponding smaller set of microorganisms may have obligate interactions, such as metabolic pathway cross-feeding and environmental functions like oxygen scavenging. The microbiome can also be subdivided into more tractable subsets via cell sorting. Microorganisms can be partitioned by size, and based on other properties using FISH to target genes of interest. This approach was used to discover and describe previously undetected, ultra-small Actinobacteria from the deepest parts of the Mediterranean (Ghai et al. 2013). In the extreme case, sorting can yield individual cells that can then be sequenced using single-cell techniques such as multiple displacement amplification (MDA). Although MDA does not produce complete genome sequences and is susceptible to contamination, the sequences obtained are informative and can serve as scaffolds for the assembly of additional metagenomic data.

Recent advances in cell culture techniques have enabled the development of “culturomic” approaches, in which microbiome samples are transferred to a range of different types of media, with differences in growth conditions favoring different microorganisms. Growing a range of lineages from the microbiome in isolation allows experimental screening and testing to be carried out, complementing the molecular analysis.

Studies that combine different types of data can highlight the role of regulatory and other processes in the microbiome. Although gene expression in bacteria is thought to be realized at

the transcriptional level, recent work has suggested an under-appreciated regulatory role for post-translational modification. Chen et al. (2016) integrated over 1000 metatranscriptomic and metaproteomic datasets to propose an inverse relationship between genome size and the importance of post-transcriptional regulation in *Mycoplasma genitalium*. McHardy et al. (2013) combined metagenomic and meta-metabolomic analysis to identify correlations between the presence of specific genera in the gut microbiome and specific metabolites observed. Among other observations, they found strong associations between *Roseburia* and *Faecalibacterium* and predicted short-chain fatty acid synthesis enzymes; given the importance of short-chain fatty acids in inflammation and immunity, and as a nutrient source for colonocytes, this finding highlighted the importance of these genera in the human gut.

Summary

The challenge of assessment and characterization of the microbiome of different habitats is clearly illustrated by the myriad approaches that have been developed to sample and analyze microbial samples. Microbiome analysis inherits all of the challenges and limitations of microbial genome analysis, then overlays the challenges of high diversity, temporal instability, and uncertain taxonomic and ecological units. Biases associated with sampling and analysis can often skew the results. However, rapid advancement in bioinformatic techniques over the last 10 years have yielded robust results, and opened up new investigations into the structure and function of the microbiome.

Anticipated improvements in microbiome sampling and analysis techniques will lead to improved understanding of microbial communities in many settings. On the technical side, long-read DNA sequencing will revolutionize the assembly and analysis of metagenomic data. Although long sequence reads will increase the robustness of marker-gene analysis as well, it remains to be seen whether 16S-based approaches retain their popularity in the coming years as shotgun sequencing methods and single-cell sequencing methods continue to become cheaper and more tractable. Growth in reference databases of genomes, including genomes assembled from metagenomic data, will help to improve the taxonomic resolution of inferred community structure. However, one of the most important shifts in the next 5 years will see the increasing adoption and integration of meta-omic techniques, as well as meta-omic datasets, to couple genetic potential with metabolic activity and explicit connections among community members. The intersection of these approaches will be an increasing area of focus for bioinformatic techniques in the near future.

Internet Resources

Major data resources

Earth Microbiome Project	Large database of marker-gene surveys	www.earthmicrobiome.org
Genomes OnLine Database (GOLD)	Database for genome projects, compliant with data-reporting standards	gold.jgi.doe.gov
Greengenes (Second Genome)	16S reference database	greengenes.secondgenome.com
Human Microbiome Project (HMP)	HMP Data Analysis Control Center	hmpdacc.org
MetaHIT	European reference metagenome project	www.metahit.eu
MG-RAST metagenomics data server	Metagenomics data and analysis server	metagenomics.anl.gov
Ribosomal Database Project	16S reference database	rdp.cme.msu.edu

SILVA	16S reference database	www.arb-silva.de
Tara Oceans	Databases for Tara Oceans expedition	www.ebi.ac.uk/services/tara-oceans-data
<i>Functional information resources</i>		
CARD	The Comprehensive Antibiotic Resistance Database	card.mcmaster.ca
CAZy	Carbohydrate metabolism database	www.cazy.org
Gene Ontology	Functional classifications of proteins, with evidence codes	www.geneontology.org
Kyoto Encyclopedia of Genes and Genomes	Large database of function, pathway, and module information	www.genome.jp/kegg
UniProtKB/Swiss-Prot	Database of experimentally validated protein functions	web.expasy.org/docs/swiss-prot_guideline.html
<i>Marker-gene analysis tools</i>		
FastQC	Quality control application for sequence reads	www.bioinformatics.babraham.ac.uk/projects/fastqc
mothur	Integrated pipeline for marker-gene analysis	www.mothur.org
PICRUSt	Software to predict metagenome function from marker genes	picrust.github.io/picrust
QIIME 2	Integrated pipeline for marker-gene analysis	qiime2.org
<i>Metagenomic analysis tools</i>		
BiomeNet	Method for identifying reactions that are characteristic of different types of metagenomes	sourceforge.net/projects/biomenet
HUMANN	Software for pathway annotation of metagenomes	huttenhower.sph.harvard.edu/humann
metAMOS	Metagenome workflow software	www.cbcb.umd.edu/software/metamos
MEtaGenome ANalyzer (MEGAN) 6	Metagenome analysis software, including last common ancestor algorithm for classification	ab.inf.uni-tuebingen.de/software/megan6
Microbiome Helper	Metagenome workflow software	github.com/mlangill/microbiome_helper

Further Reading

- Franzosa, E.A., Hsu, T., Sirota-Madi, A. et al. (2015). Sequencing and beyond: integrating molecular 'omics' for microbial community profiling. *Nat. Rev. Microbiol.* 13: 360–372. A review and prospectus of emerging DNA-based and complementary meta-omic approaches, and their implications for future microbiome studies.
- Hanage, W.P. (2014). Microbiome science needs a healthy dose of scepticism. *Nature* 512: 247. A short piece that outlines some of the key pitfalls in the interpretation of metagenomic data and the identification of biologically meaningful trends in the data.
- Sczyrba, A., Hofmann, P., Belmann, P. et al. (2017). Critical assessment of metagenome interpretation – a benchmark of metagenomics software. *Nat. Methods* 14: 1063. A recent comparative evaluation of different techniques to assemble and assign taxonomic information to metagenomic data.
- Sharpton, T.J. (2014). An introduction to the analysis of shotgun metagenomic data. *Front. Plant Sci.* 5: 209. A review of analytical techniques for metagenome analysis that lists and cites a wide range of approaches.

References

- Abubucker, S., Segata, N., Goll, J. et al. (2012). Metabolic reconstruction for metagenomic data and its application to the human microbiome. *PLoS Comput. Biol.* 8: e1002358.
- Albertsen, M., Hugenholtz, P., Skarshewski, A. et al. (2013). Genome sequences of rare, uncultured bacteria obtained by differential coverage binning of multiple metagenomes. *Nat. Biotechnol.* 31: 533–538.
- Altschul, S.F., Madden, T.L., Schäffer, A.A. et al. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* 25: 3389–3402.
- Baas-Becking, L.G.M. (1934). *Geobiologie; of inleiding tot de milieukunde*. [In Dutch.]. The Hague, Netherlands: WP Van Stockum & Zoon NV.
- Baichoo, S. and Ouzounis, C.A. (2017). Computational complexity of algorithms for sequence comparison, short-read assembly and genome alignment. *Biosystems* 156: 72–85.
- Balvočiūtė, M. and Huson, D.H. (2017). SILVA, RDP, Greengenes, NCBI and OTT—how do these taxonomies compare? *BMC Genomics* 18: 114.
- Berry, D., Stecher, B., Schintlmeister, A. et al. (2013). Host-compound foraging by intestinal microbiota revealed by single-cell stable isotope probing. *Proc. Natl. Acad. Sci. USA.* 110: 4720–4725.
- Boisvert, S., Raymond, F., Godzaridis, É. et al. (2012). Ray Meta: scalable de novo metagenome assembly and profiling. *Genome Biol.* 13: R122.
- Brown, M.V., Lauro, F.M., DeMaere, M.Z. et al. (2012). Global biogeography of SAR11 marine bacteria. *Mol. Syst. Biol.* 8: 595.
- Buchfink, B., Xie, C., and Huson, D.H. (2015). Fast and sensitive protein alignment using DIAMOND. *Nat. Methods* 12: 59–60.
- Callahan, B.J., McMurdie, P.J., Rosen, M.J. et al. (2016). DADA2: high-resolution sample inference from Illumina amplicon data. *Nat. Methods* 13: 581.
- Cantarel, B.L., Coutinho, P.M., Rancurel, C. et al. (2008). The Carbohydrate-Active EnZymes database (CAZy): an expert resource for glycogenomics. *Nucleic Acids Res.* 37 (Database issue): D233–D238.
- Caporaso, J.G., Kuczynski, J., Stombaugh, J. et al. (2010). QIIME allows analysis of high-throughput community sequencing data. *Nat. Methods* 7: 335–336.
- Caporaso, J.G., Paszkiewicz, K., Field, D. et al. (2012). The Western English Channel contains a persistent microbial seed bank. *ISME J.* 6: 1089–1093.
- Chen, W.H., van Noort, V., Lluch-Senar, M. et al. (2016). Integration of multi-omics data of a genome-reduced bacterium: prevalence of post-transcriptional regulation and its correlation with protein abundances. *Nucleic Acids Res.* 44: 1192–1202.
- Choo, J.M., Leong, L.E., and Rogers, G.B. (2015). Sample storage conditions significantly influence faecal microbiome profiles. *Sci. Rep.* 5: 16350.
- Cock, P.J., Fields, C.J., Goto, N. et al. (2009). The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Res.* 38: 767–1771.
- Cole, J.R., Wang, Q., Fish, J.A. et al. (2014). Ribosomal Database Project: data and tools for high throughput rRNA analysis. *Nucleic Acids Res.* 42 (Database issue): D633–D642.
- Comeau, A.M., Douglas, G.M., and Langille, M.G. (2017). Microbiome Helper: a custom and streamlined workflow for microbiome research. *mSystems* 2: e00127–e00116.
- Darling, A.E., Jospin, G., Lowe, E. et al. (2014). PhyloSift: phylogenetic analysis of genomes and metagenomes. *PeerJ* 2: e243.
- DeSantis, T.Z., Hugenholtz, P., Larsen, N. et al. (2006). Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB. *Appl. Environ. Microbiol.* 72: 5069–5072.
- Edgar, R.C., Haas, B.J., Clemente, J.C. et al. (2011). UCHIME improves sensitivity and speed of chimera detection. *Bioinformatics* 27: 2194–2200.
- Ghai, R., Mizuno, C.M., Picazo, A. et al. (2013). Metagenomics uncovers a new group of low GC and ultra-small marine Actinobacteria. *Sci. Rep.* 3: 2471.

- Giovannoni, S.J. (2017). SAR11 bacteria: the most abundant plankton in the oceans. *Annu. Rev. Marine Sci.* 9: 231–255.
- Giovannoni, S.J., Britschgi, T.B., Moyer, C.L., and Field, K.G. (1990). Genetic diversity in Sargasso Sea bacterioplankton. *Nature* 345: 60.
- Hanage, W.P. (2014). Microbiome science needs a healthy dose of scepticism. *Nature* 512: 247.
- Hamady, M., Walker, J.J., Harris, J.K. et al. (2008). Error-correcting barcoded primers for pyrosequencing hundreds of samples in multiplex. *Nat. Methods* 5: 235.
- Handelsman, J., Rondon, M.R., Brady, S.F. et al. (1998). Molecular biological access to the chemistry of unknown soil microbes: a new frontier for natural products. *Chem. Biol.* 5: R245–R249.
- Hird, S.M., Carstens, B.C., Cardiff, S.W. et al. (2014). Sampling locality is more detectable than taxonomy or ecology in the gut microbiota of the brood-parasitic Brown-headed Cowbird (*Molothrus ater*). *PeerJ* 2: e321.
- Hunt, D.E., Klepac-Ceraj, V., Acinas, S.G. et al. (2006). Evaluation of 23S rRNA PCR primers for use in phylogenetic studies of bacterial diversity. *Appl. Environ. Microbiol.* 72: 2221–2225.
- Huson, D.H., Beier, S., Flade, I. et al. (2016). MEGAN community edition-interactive exploration and analysis of large-scale microbiome sequencing data. *PLoS Comput. Biol.* 12: e1004957.
- Huttenhower, C., Gevers, D., Knight, R. et al. (2012). Structure, function and diversity of the healthy human microbiome. *Nature* 486: 207.
- Jia, B., Raphenya, A.R., Alcock, B. et al. (2017). CARD 2017: expansion and model-centric curation of the comprehensive antibiotic resistance database. *Nucleic Acids Res.* 45 (D1): D566–D573.
- Jonsson, V., Österlund, T., Nerman, O., and Kristiansson, E. (2016). Statistical evaluation of methods for identification of differentially abundant genes in comparative metagenomics. *BMC Genomics* 17: 78.
- Kanehisa, M., Furumichi, M., Tanabe, M. et al. (2017). KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res.* 45 (D1): D353–D361.
- Kang, D.W., Adams, J.B., Gregory, A.C. et al. (2017). Microbiota transfer therapy alters gut ecosystem and improves gastrointestinal and autism symptoms: an open-label study. *Microbiome* 5: 10.
- Karlsson, E.A., Small, C.T., Freiden, P. et al. (2015). Non-human primates harbor diverse mammalian and avian astroviruses including those associated with human infections. *PLoS Pathog.* 11: e1005225.
- Kembel, S.W., Wu, M., Eisen, J.A., and Green, J.L. (2012). Incorporating 16S gene copy number information improves estimates of microbial diversity and abundance. *PLoS Comput. Biol.* 8: e1002743.
- Knights, D., Costello, E.K., and Knight, R. (2011). Supervised classification of human microbiota. *FEMS Microbiol. Rev.* 35: 343–359.
- Koskinen, K., Pausan, M.R., Perras, A.K. et al. (2017). First insights into the diverse human archaeome: specific detection of archaea in the gastrointestinal tract, lung, and nose and on skin. *MBio* 8: e00824–e00817.
- Labrière, N., Laumonier, Y., Locatelli, B. et al. (2015). Ecosystem services and biodiversity in a rapidly transforming landscape in Northern Borneo. *PLoS One* 10: e0140423.
- Langille, M.G., Zaneveld, J., Caporaso, J.G. et al. (2013). Predictive functional profiling of microbial communities using 16S rRNA marker gene sequences. *Nat. Biotechnol.* 31: 814–821.
- Langmead, B. and Salzberg, S.L. (2012). Fast gapped-read alignment with Bowtie 2. *Nat. Methods* 9: 357–359.
- Lee, S.C., San Tang, M., Lim, Y.A. et al. (2014). Helminth colonization is associated with increased diversity of the gut microbiota. *PLoS Negl. Trop. Dis.* 8: e2880.
- Ley, R.E., Bäckhed, F., Turnbaugh, P. et al. (2005). Obesity alters gut microbial ecology. *Proc. Natl. Acad. Sci. USA.* 102: 11070–11075.
- Li, H. and Durbin, R. (2009). Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics* 25: 1754–1760.

- Love, M.I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.* 15: 550.
- Lozupone, C. and Knight, R. (2005). UniFrac: a new phylogenetic method for comparing microbial communities. *Appl. Environ. Microbiol.* 71: 8228–8235.
- Lu, Z., Deng, Y., Van Nostrand, J.D. et al. (2012). Microbial gene functions enriched in the Deepwater Horizon deep-sea oil plume. *ISME J.* 6: 451–460.
- Mahé, F., Rognes, T., Quince, C. et al. (2014). Swarm: robust and fast clustering method for amplicon-based studies. *PeerJ* 2: e593.
- Matsen, F.A., Kodner, R.B., and Armbrust, E.V. (2010). pplacer: linear time maximum-likelihood and Bayesian phylogenetic placement of sequences onto a fixed reference tree. *BMC Bioinformatics.* 11: 538.
- McHardy, I.H., Goudarzi, M., Tong, M. et al. (2013). Integrative analysis of the microbiome and metabolome of the human intestinal mucosal surface reveals exquisite inter-relationships. *Microbiome* 1: 17.
- Murat Eren, A.M., Maignien, L., Sul, W.J. et al. (2013). Oligotyping: differentiating between closely related microbial taxa using 16S rRNA gene data. *Meth. Ecol. Evol.* 4: 1111–1119.
- NCBI Resource Coordinators (2018). Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.* 46 (D1): D8–D13.
- Ning, J. and Beiko, R.G. (2015). Phylogenetic approaches to microbial community classification. *Microbiome* 3: 47.
- Nurk, S., Meleshko, D., Korobeynikov, A., and Pevzner, P.A. (2017). metaSPAdes: a new versatile metagenomic assembler. *Genome Res.* 27: 824–834.
- Parks, D.H. and Beiko, R.G. (2013). Measures of phylogenetic differentiation provide robust and complementary insights into microbial communities. *ISME J.* 7: 173–183.
- Parks, D.H., Tyson, G.W., Hugenholtz, P., and Beiko, R.G. (2014). STAMP: statistical analysis of taxonomic and functional profiles. *Bioinformatics* 30: 3123–3124.
- Paulson, J.N., Stine, O.C., Bravo, H.C., and Pop, M. (2013). Differential abundance analysis for microbial marker-gene surveys. *Nat. Methods* 10: 1200–1202.
- Prestat, E., David, M.M., Hultman, J. et al. (2014). FOAM (functional ontology assignments for metagenomes): a hidden Markov model (HMM) database with environmental focus. *Nucleic Acids Res.* 42: e145–e145.
- Quast, C., Pruesse, E., Yilmaz, P. et al. (2013). The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. *Nucleic Acids Res.* 41 (Database issue): D590–D596.
- Robinson, M.D., McCarthy, D.J., and Smyth, G.K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* 26: 139–140.
- Rocap, G., Distel, D.L., Waterbury, J.B., and Chisholm, S.W. (2002). Resolution of *Prochlorococcus* and *Synechococcus* ecotypes by using 16S-23S ribosomal DNA internal transcribed spacer sequences. *Appl. Environ. Microbiol.* 68: 1180–1191.
- Rosen, G.L., Reichenberger, E.R., and Rosenfeld, A.M. (2011). NBC: the Naive Bayes Classification tool webserver for taxonomic classification of metagenomic reads. *Bioinformatics* 27: 127–129.
- Schirmer, M., Ijaz, U.Z., D’Amore, R. et al. (2015). Insight into biases and sequencing errors for amplicon sequencing with the Illumina MiSeq platform. *Nucleic Acids Res.* 43: e37.
- Schloss, P.D., Westcott, S.L., Ryabin, T. et al. (2009). Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Appl. Environ. Microbiol.* 75: 7537–7541.
- Seedorf, H., Kittelmann, S., Henderson, G., and Janssen, P.H. (2014). RIM-DB: a taxonomic framework for community structure analysis of methanogenic archaea from the rumen and other intestinal environments. *PeerJ.* 2: e494.
- Segata, N., Izard, J., Waldron, L. et al. (2011). Metagenomic biomarker discovery and explanation. *Genome Biol.* 12: R60.
- Segata, N., Waldron, L., Ballarini, A. et al. (2012). Metagenomic microbial community profiling using unique clade-specific marker genes. *Nat. Methods* 9: 811–814.

- Shade, A., Jones, S.E., Caporaso, J.G. et al. (2014). Conditionally rare taxa disproportionately contribute to temporal changes in microbial diversity. *MBio* 5: e01371–e01314.
- Shafiei, M., Dunn, K.A., Chipman, H. et al. (2014). BiomeNet: a Bayesian model for inference of metabolic divergence among microbial communities. *PLoS Comput. Biol.* 10: e1003918.
- Smith, C.A., Want, E.J., O’Maille, G. et al. (2006). XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. *Anal. Chem.* 78: 779–787.
- Sonnenborn, U. and Schulze, J. (2009). The non-pathogenic *Escherichia coli* strain Nissle 1917—features of a versatile probiotic. *Microb. Ecol. Health Dis.* 21: 22–158.
- Stackebrandt, E. and Goebel, B.M. (1994). Taxonomic note: a place for DNA-DNA reassociation and 16S rRNA sequence analysis in the present species definition in bacteriology. *Int. J. Syst. Evol. Microbiol.* 44: 846–849.
- Stahl, D.A., Lane, D.J., Olsen, G.J., and Pace, N.R. (1985). Characterization of a Yellowstone hot spring microbial community by 5S rRNA sequences. *Appl. Environ. Microbiol.* 49: 1379–1384.
- Stearns, J.C., Lynch, M.D., Senadheera, D.B. et al. (2011). Bacterial biogeography of the human digestive tract. *Sci. Rep.* 1: 170.
- Sunagawa, S., Coelho, L.P., Chaffron, S. et al. (2015). Structure and function of the global ocean microbiome. *Science* 348: 1261359.
- Thompson, L.R., Sanders, J.G., McDonald, D. et al. (2017). A communal catalogue reveals Earth’s multiscale microbial diversity. *Nature* 551: 457–463.
- Treangen, T.J., Koren, S., Sommer, D.D. et al. (2013). MetAMOS: a modular and open source metagenomic assembly and analysis pipeline. *Genome Biol.* 14: R2.
- Turnbaugh, P.J., Ley, R.E., Hamady, M. et al. (2007). The human microbiome project. *Nature* 449: 804.
- Tyson, G.W., Chapman, J., Hugenholtz, P. et al. (2004). Community structure and metabolism through reconstruction of microbial genomes from the environment. *Nature* 428: 37.
- UniProt Consortium (2018). UniProt: the universal protein knowledgebase. *Nucleic Acids Res.* 46: 2699.
- Venter, J.C., Remington, K., Heidelberg, J.F. et al. (2004). Environmental genome shotgun sequencing of the Sargasso Sea. *Science* 304: 66–74.
- Vuong, H.E. and Hsiao, E.Y. (2017). Emerging roles for the gut microbiome in autism spectrum disorder. *Biol. Psychiatry* 81: 411–423.
- Wang, Q., Garrity, G.M., Tiedje, J.M., and Cole, J.R. (2007). Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. *Appl. Environ. Microbiol.* 73: 5261–5267.
- Weiss, S., Xu, Z.Z., Peddada, S. et al. (2017). Normalization and microbial differential abundance strategies depend upon data characteristics. *Microbiome* 5: 27.
- Wood, D.E. and Salzberg, S.L. (2014). Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biol.* 15: R46.
- Ye, Y. and Doak, T.G. (2009). A parsimony approach to biological pathway reconstruction/inference for genomes and metagenomes. *PLoS Comput. Biol.* 5: e1000465.
- Zhang, J., Kobert, K., Flouri, T., and Stamatakis, A. (2014). PEAR: a fast and accurate Illumina Paired-End reAd mergeR. *Bioinformatics* 30: 614–620.
- Zhu, W., Lomsadze, A., and Borodovsky, M. (2010). Ab initio gene identification in metagenomic sequences. *Nucleic Acids Res.* 38: e132–e132.

17

Translational Bioinformatics*Sean D. Mooney and Stephen J. Mooney***Introduction**

Translational bioinformatics is an emergent discipline that leverages informatics and computational methods to bridge the basic sciences with the clinic and clinical sciences. With advances in genomic and other high-throughput technologies, we are seeing a revolution surrounding informatics methods that directly impact human health. Given that we can now inexpensively and rapidly profile human tissues and biofluids with highly reproducible precision, this exciting field is growing quickly and evolving even faster. This chapter outlines the basics of translational informatics, as well as areas of active research. The field is driven by our ability to collect vast quantities of heterogeneous data, all describing patients, their environment, and various experimental models of patient phenotype, including those deduced using model organisms. Translational informaticians develop or leverage computational tools to deepen our understanding of the relationship between genotype and phenotype, then use that knowledge to improve patient health outcomes.

Technologies and data that are relevant to translational bioinformatics include high-throughput molecular data (such as genomes, transcriptomes, proteomes, and metabolomes), electronic health record (EHR) data, behavioral data such as those derived from smartphone sensors, social media data, and environmental exposure data such as air or water quality measurements. Computational tools are actively being developed to integrate two or more of these types of data and then use them to describe and understand patient phenotypes. From a phenotypic model, predictions can then be made about intervention outcomes, such as treatment response or disease progression. For example, researchers are working on blood tests for cancer detection, tests for clinical depression risk based on mobile smartphone activities, and early detection of kidney disease from urine. However, much work still needs to be done, as methods developed for human genomes and to identify disease risk variants are still not as accurate as clinicians themselves. Similarly, new methods for data integration are being developed that include new types of data, making data integration easier through the use of standard data models. Large groups of patient research study participants (cohorts) are being recruited to understand population-based risks of disease and treatment outcomes, creating even more data analysis opportunities; these projects include the Million Veterans Program, the National Institute of Health's (NIH) All of Us initiative, and the UK Biobank.

Understanding and performing computational analyses using patient phenotypes is becoming increasingly important in the study of genes and proteins, and that is one of the major foci of the field of translational informatics. As we begin to understand the molecular causes of any given disease at the gene and protein level, we begin to understand that differences in the actual clinical expression of that disease can have a molecular basis. Quantifying a patient's individual phenotype to tailor treatment is the basis of an important area of active research called "precision medicine." Precision medicine also helps us understand the molecular causes of

a disease better, by developing increasingly precise classifications of patients into groups or subtypes, thereby reducing heterogeneity of clinical observations.

To that end, the promise of translational informatics is very high. It is one of the few fields of bioinformatics that can be directly used in the context of patient care. Methods to identify pathogenic variants can be used by clinical geneticists to make recommendations about whether a patient is likely to be susceptible to a genetic disease. Tools for repurposing drugs or drug interactions can be used by clinicians to identify custom therapies for patients. Biochemical markers of subtype and disease staging can be used to make treatment decisions in cancer. Methods that predict risks of disease can be integrated directly into the patient's chart and the health provider's clinical informatics system. However, these applications do not come without risk. The collection of large heterogeneous datasets describing patients can have an impact on patient privacy. Similarly, inaccurate or poorly developed methods applied over-confidently can lead to patient harm. It is important to understand the risks, benefits, and appropriate application of any translational informatics study or method.

Databases Describing the Genetics of Human Health

One of the central challenges in the field of translational bioinformatics is understanding the relationship between genotype and phenotype. As a result, much of the early focus of this field centered on the collection and interpretation of human genomes. Sequencing genomes in clinical settings is increasingly widespread and is being applied to many more patient populations than before. This has given rise to large databases of genetic variants that have been annotated with human diseases and phenotypes. These databases are based on scientific findings from studies that identify genetic variants that are found in patients with certain conditions or phenotypes more often than one would expect by chance; these studies are called *association studies*. As association studies are based on a statistical observation, they may not necessarily identify the underlying genetic cause of a disease, particularly if the number of participants in the study is relatively small. Early databases of genetic variants associated with specific diseases were created by manually curating papers that described identified pathogenic variants, but we have since learned that some of these variants were incorrectly associated and are not actually pathogenic. To overcome this, newer genetic variant resources include not only evidence from peer-reviewed publications but also provide other lines of evidence of pathogenicity and causation, including genetic testing facility reports. The American College of Medical Genetics (ACMG) has recently developed guidelines for the clinical interpretation of genetic variants that includes the use of computational methods (Richards et al. 2015).

For largely historical reasons, pathogenic genetic variants are not located in a single, central database. To fully annotate genomes with possible pathogenic variants, multiple resources must still be used. There are ongoing efforts to create integrated interfaces that provide access to many databases centrally, such as MyVariant.info (Xin et al. 2016), but these resources generally do not provide access to the unique and extensive kinds of data found within commercial or “commercial-like” genetic disease databases that require a license for use. For variants that cause inherited disorders, databases such as the commercial Human Gene Mutation Database (HGMD) (Stenson et al. 2017) or the freely available ClinVar database (Landrum et al. 2016) provide curated information on pathogenic variants and are widely used to annotate sequenced human genomes. (Figure 17.1 includes a screenshot of a ClinVar entry for a variant in the *CFTR* gene, the gene responsible for cystic fibrosis.) Variants that are associated with differences in treatment response are curated in the Pharmacogenetics Knowledge-Base (PharmGKB) (Thorn et al. 2010) and in DrugBank (Wishart et al. 2018), and clinical guidelines for the use of those variants are provided by the Clinical Pharmacogenetics Implementation Consortium (CPIC) (Relling and Klein 2011). Pharmacogenetics examples include codeine, which is metabolized by the *CYP2D6* product, and warfarin, which is metabolized by *VKORC1* and *CYP2C9* products; in both cases, variants in these genes are associated with different responses to these drugs. Somatic variants found in tumors are collected in the COSMIC database (Forbes et al. 2017). Other databases of interest include the publicly curated wiki

NCBI Resources How To ClinVar

Search ClinVar for gene symbols, HGVS expressions, conditions, and more

Advanced

Home About Access Help Submit Statistics FTP

NM_000492.3(CFTR):c.-8G>C

Variation ID: 93148
 Review status: ★ ★ ★ criteria provided, multiple submitters, no conflicts

1 Affected gene
 cystic fibrosis transmembrane conductance regulator (CFTR) [Gene - OMIM - Variation Viewer]
 Search ClinVar for variants within CFTR
 Search ClinVar for variants including CFTR

Interpretation Go to:

Clinical significance: [Benign/likely benign](#)
 Last evaluated: Jun 14, 2016
 Number of submission(s): 4
 Condition(s): Cystic fibrosis [MedGen - Orphanet - OMIM]
[See supporting ClinVar records](#)

Allele(s) Go to:

NM_000492.3(CFTR):c.-8G>C

Allele ID: 99055
 Variant type: single nucleotide variant
 Cytogenetic location: 7q31.2
 Genomic location: Chr7: 117480087 (on Assembly GRCh38)
 Chr7: 117120141 (on Assembly GRCh37)
 HGVS: NG_016465.4.g.19304G>C
 NM_000492.3.c.-8G>C
 NC_000007.14.g.117480087G>C (GRCh38)
 NC_000007.13.g.117120141G>C (GRCh37)
 Links: dbSNP: [1800501](#)
 NCBI 1000 Genomes Browser: [rs1800501](#)
 Molecular consequence: NM_000492.3:c.-8G>C: 5 prime UTR variant [Sequence Ontology [SO:0001623](#)]
 Allele frequency: GO-ESP 0.04060 (C)
 GMAF 0.02740 (C)
 ExAC 0.04549 (C)

Variant frequency in dbGaP Go to:

NM_000492.3(CFTR):c.-8G>C
 GRCh37 Chr7:117120141

	Called variants	Potential variants
Sample count	15748 of 24193	3656 of 40268

Called variants are samples submitted to dbGaP that have the variant allele. Potential variants are SRA runs that display the allele in at least 30% of the reads covering the position, and have 10 or more passing reads covering the position.

Browser views Go to:

RefSeqGene
 Variation Viewer [GRCh38 - GRCh37]
 UCSC [GRCh38/hg38 - GRCh37/hg19]

Related information Go to:

dbSNP
 Functional Class
 Gene
 MedGen
 OMIM
 PMC
 PubMed
 Related genes (specific)

Assertion and evidence details Go to:

Clinical assertions Summary evidence Supporting observations

Germline Filter:

Clinical significance (Last evaluated)	Review status (Assertion method)	Collection method	Condition(s) (Mode of inheritance)	Origin	Citations	Submitter - Study name	Submission accession
Benign (Mar 27, 2015)	criteria provided, single submitter - EGL Classification Definitions	clinical testing	not specified [MedGen]	germline	PubMed (1) [See all records that cite this PMID]	Emory Genetics Laboratory, Emory University	SCV000110839.6
Benign (Feb 21, 2013)	criteria provided, single submitter - LMM Criteria	clinical testing	not specified [MedGen]	germline		Laboratory for Molecular Medicine Partners HealthCare Personalized Medicine	SCV000197432.3
Benign	criteria provided, single submitter - ACMG Guidelines 2015 - ACMG Guidelines 2015	clinical testing	not specified [MedGen]	germline		PreventionGenetics, PreventionGenetics	SCV000304468.1
Likely benign (Jun 14, 2016)	criteria provided, single submitter - ICSL Variant Classification 20161018	clinical testing	Cystic fibrosis [MedGen Orphanet OMIM]	germline		Illumina Clinical Services Laboratory, Illumina	SCV000466501.2

Figure 17.1 ClinVar entry for a benign variant in the cystic fibrosis gene (CFTR). Variant includes multiple reports from testing centers of evidence indicating the single nucleotide variant is non-pathogenic.

SNPedia (Cariaso and Lennon 2012), a resource of curated human single nucleotide polymorphisms. Interestingly, the developers of SNPedia have developed a whole genome analysis tool, Promethease, which uses SNPedia to annotate sequenced whole genomes or exomes. Another highly used tool for annotating sequenced human genomes is ANNOVAR, which takes Variant Call Format (or VCF, a standard text file format for annotated genetic variants) files as input (Wang et al. 2010) and maps variants to genes, identifies classes of variants (such as insertions, deletions, and single nucleotide variants), and provides other useful annotations, such as mutation impact predictions.

Prediction and Characterization of Impactful Genetic Variants from Sequence

Traditionally, associations of genetic variants with human disease were determined using statistical differences in genetic markers inferred in case–control studies (Collins et al. 1999). Recently, many complementary algorithms that use our understanding of biology to predict and understand genetic variants that cause disease have been published. (For a review, please see Cooper and Shendure [2011].) Variants can include single nucleotide variants, short insertions and deletions (referred to as “indels” that are generally less than a few thousand bases), and larger insertions and deletions that cause structural variations that can be quite large and contain entire genes. Genetic variants that are discovered in the human population can be grouped into categories based on whether they are relevant to human diseases and conditions (Box 17.1). They include pathogenic variants, or variants that cause disease. Variants of unknown significance (VUS) possibly cause disease. Polymorphisms are present in the population but do not cause a disease or condition. Bioinformaticians have known for some time that pathogenic genetic variants tend to occur at sites of evolutionary pressure (Mooney and Klein 2002). With the discovery that sites of functional and evolutionary importance are more likely to be mutated in human genetic disease and the concurrent growth of databases of known pathogenic variants, computational methods have been developed that use both to predict new genetic variants that are impactful. Typically, these approaches are *supervised* – that is, they use a database of known impactful variants (having an effect) and neutral variants (having no effect) to determine whether a variant that has never been seen before is likely to be in the impactful set or the neutral set. These methods require a database of annotated variants (a *training set*), statistically useful proteomic and genomic features for classification, and a classification approach for prediction that uses the training set and features to classify variants.

Box 17.1 Gene Testing of Hereditary Cancers

Cancer risk can be significantly increased and age of onset can be significantly reduced, depending on which genetic variants in specific genes an individual has inherited from their parents. These cancers include specific types of breast, ovarian, colorectal, and prostate cancers. Patients that inherit risk of these “hereditary cancers” are important to identify early, as preventive screening or procedures (such as mastectomies) can be performed in order to minimize cancer impact. In order to test for specific variants in risk genes such as *BRCA1*, *BRCA2*, *TP53*, and *PTEN*, genetic tests can be performed. Color Genomics, Invitae, Myriad Genetics, and other companies provide tests that can be ordered by a provider and by the patient themselves. These test results provide risk assessments and can identify both pathogenic and variants of unknown significance.

Characterizing Genetic Variants at the Protein Level

Pathogenic variants tend to occur non-randomly in protein structures and tend to be buried within the protein, thereby disrupting protein structure (Wang and Moult 2003). These variants also tend to damage functional sites in proteins, such as binding sites (Lugo-Martinez

et al. 2016). From these observations, two types of pathogenic prediction methods have been developed. The methods differ in the training sets that are used to build them. The first group of methods use mutations that have been interrogated experimentally (generally in vitro or in a model organism). Examples of these methods include SIFT (Ng and Henikoff 2003), SNAP (Bromberg et al. 2008), and others. The second group of methods use data on human disease mutations. Examples of these methods include MutPred (Li et al. 2009), PolyPhen-2 (Adzhubei et al. 2013), and others. Although these two families of methods are similar, the fact that they use different training sets can lead to differences in the predictions they generate. The first predicts impact on protein function, while the second predicts human pathogenicity. The accuracy of these methods ranges from 65% to 85% for non-synonymous single nucleotide variants, depending on the specific software method used or even the gene where the variant occurs (Ioannidis et al. 2016). In order to make analysis of amino acid substitutions in the human proteome easier, researchers have developed the dbNSFP annotation database; this database includes annotations from many different prediction algorithms, documenting the predicted impact of all possible missense changes (Liu et al. 2016). dbNSFP contains all possible amino acid mutations in the human proteome, so investigators that discover novel mutations can look them up in the database without having to compute the impact prediction from many tools – a long and difficult task!

Characterizing Genetic Variants at the Genomic or Transcriptomic Level

Not surprisingly, genetic variants that do not directly impact the protein sequence encoded by a particular gene of interest can also cause disease. Variants can affect messenger RNA (mRNA) splicing and processing (Wang et al. 2008; Mort et al. 2014), the regulation of transcription or translation (Ritchie et al. 2014), functional sites in untranslated regions of transcripts, and sites of epigenetic importance. In a mechanism that is similar to how variants affect protein sequence and structure, variants that disrupt non-coding sites can also cause disease. While many efforts have been made to identify these variants, their discovery has lagged behind that of the more easily characterized variants that affect protein sequence. With the advent of whole genome sequencing, identifying non-coding variants of importance has greatly facilitated the identification of new pathogenic variants (Boycott et al. 2013).

Using Informatics to Prioritize Disease-Causing Genes

In addition to methods available to prioritize potential pathogenic variants, there are similar supervised approaches to prioritize genes that may cause (or be associated with) certain diseases. For example, if 15 genes were hypothetically known to cause familial forms of Parkinson disease with high certainty, what would be the likelihood of identifying a 16th causative gene? Answering this question is difficult, as there are 20 000 or so protein-coding genes in the human genome that must be sifted through quite carefully to identify actual causative genes. The hypothesis relies on each of the known training set genes sharing some molecular signatures (such as participation in common metabolic pathways) that can be captured as “features” (variables used in machine learning); these features can then be used to implicate additional genes as causative using either supervised or semi-supervised methods. This can be done using bioinformatic methods and features that are similar to methods and features used for predicting Gene Ontology terms or other annotations for a gene or protein (essentially treating a disease association as an annotation). (See also Chapter 7 and Radivojac et al. [2013] for a list of methods.) Features may include gene product function, pathways, tissue expression, shared domains, literature co-occurrence, and many others. Methods that use this “guilt by association” approach include ENDEAVOUR (Tranchevent et al. 2016), PhenoPred (Radivojac et al. 2008), and GeneMANIA (Warde-Farley et al. 2010); they have all been used for this purpose and are complementary to genetic approaches for associating new genes to traits using statistical associations.

Translating Model Organism Data to Humans

There are many humans with clinical phenotypes that are either difficult to diagnose or completely undiagnosable at the present time. Understanding these “undiagnosable diseases” is an active area for the application of clinical genome sequencing to discover novel pathogenic variants and underlying causation. Similarly, many genetic diseases have been characterized in model organisms through RNA interference screens or genetic knockouts of genes orthologous to disease genes in humans, and it is thought that these models could give insight into previously undiagnosed conditions in humans. Projects such as the Monarch Initiative (Mungall et al. 2017) are using bioinformatic approaches to facilitate translating these animal model phenotypes to human conditions and vice versa. Monarch is a broad knowledge base that is integrating heterogeneous data from literature to help interrogate these underlying causes of genetic disease, to understand patient phenotypes that are not clearly diagnosable, and to better understand the mechanisms of disease using an open science approach. (See also Informatics and Precision Medicine for a discussion of community challenges on how open data can help advance science.)

Computing with Patient Phenotype Using Data in Electronic Health Records

Introduction to Electronic Health Records

As mentioned above, describing a patient’s phenotype is becoming increasingly important for understanding the molecular causes of disease and, to that end, researchers have increasingly turned to the patient’s medical chart for insight into their phenotype (Pathak et al. 2013). Over the past 20 years, medical records data, including patient charts, are being digitized into EHRs, and the rate of adoption (and success in implementation) of these EHRs has differed from health system to health system (Jha et al. 2009). These complex systems manage all aspects of the inpatient and outpatient experience: scheduling patient visits, managing billing, submitting orders, tracking laboratory test results, facilitating return of results through patient portals, and pretty much everything else surrounding clinical care. Not surprisingly, commercial vendor EHR systems are large and complex. Unlike most other datasets described in this book, health record data were not collected for research purposes and research use is generally a secondary use that was not necessarily considered when data were collected – and, thus, health record data may be difficult to access and use (Vuokko et al. 2015). Typically, data within an EHR can be structured (coded) or unstructured (such as plaintext or images). Structured data use specific terminologies, data dictionaries, or numerical values that can be easily analyzed. Today, as much as 80% of the data within an EHR system are unstructured, thereby creating a research challenge for bioinformaticians in tapping the heretofore inaccessible phenotypic information found within these patient records.

Data stored in EHR systems can follow data models that are unique to the site collecting the data, even when a common vendor system is used. In order to extract the data from the source system or to integrate with other systems data, a process of extraction, transform, and loading of the data must be performed for them to be loaded into a clinical data repository that can be analyzed. The extraction step downloads the data from the source system. The transformation step may require changes in the data model or mapping specific fields to new terminologies or datatypes. Data abstraction in the transformation step can be costly and may involve significant manual curation. Throughout this process, data quality must be assessed and, if possible, data must be corrected to maintain the highest standards. Finally, the loading process integrates the transformed data into a data repository or data warehouse that houses the integrated datasets. These repositories are generally relational databases and can be queried readily using the widely used Structured Query Language (SQL).

Table 17.1 Examples of commonly used biomedical ontologies and terminologies in translational research.

Ontology	Description
Gene Ontology (GO)	Three terminologies widely used for annotating genes and proteins describing molecular functions, biological processes, and cellular components.
Human Phenotype Ontology (HPO)	Concepts describing human phenotype and disease.
International Classification of Diseases (ICD)	Diagnosis codes widely used in electronic health record systems.
Medical Subject Headings (MeSH)	Terms used largely by librarians and curators to annotate and categorize biomedical literature.
National Drug File (NDF)	Standardized name, dosing and strength, package size, National Drug Code, and other metadata for drugs in use at the Veterans Affairs Medical Centers.
Phenotypic Quality Ontology (PATO)	Terminology for annotating model organism phenotypes.
RxNORM	Normalized drug names linked to many other terminologies.
Systematized Nomenclature of Medicine – Clinical Terms (SNOMEDCT)	A widely used clinical ontology.

Structured Clinical Data with Biomedical Ontologies

The development of biomedical ontologies has been critical in enabling the ability to effectively mine and analyze patient data. An ontology is a structured vocabulary that describes a domain with semantic relationships between the terms, and there are resources online that provide access to standardized ontologies such as those provided by the National Center for Biomedical Ontology (NCBO) (Musen et al. 2012) or the Open Biomedical Ontologies Consortium (OBO) (Smith et al. 2007). Ontologies provide standard terminologies or codes that can be used for describing a patient's health status. For example, the International Classification of Diseases (ICD) is likely the most used ontology in the world and is used as the standard diagnosis, problem, and billing code classification for health record systems (Anonymous 1996). While ICD only provides diagnosis codes, the Human Phenotype Ontology (HPO) provides access to concepts of human phenotype (Kohler et al. 2017). For example, a patient may be diagnosed with influenza (ICD terms under the identifier ICD:J09) in the EHR, but the patient may have the phenotype of vomiting (HPO identifier HP:0002013), which might not have been entered into the health record. Table 17.1 provides examples of terminologies used for coding in translational informatics. Often, vendor systems will use data dictionaries that are not standardized, requiring subsequent mapping to a standard ontology to be useful for research purposes.

Common Data Models

As clinics, hospitals, and health systems accumulate more and more clinical data derived from their patients, interoperability of those data with other datasets becomes important. Interoperable data are useful operationally for exchanging data across different clinical sites. They are also useful for research using data derived from different sources, such as comparative effectiveness research, discovery of new risks of diseases or conditions, epidemiological data science, and more. Data interoperability is achieved through common data models that allow for integration and comparison. Common data models define standard tables and standard ontologies that are used to describe clinical data. These data models will describe patient information and demographics, patient encounters (or visits) including vitals and billing diagnosis codes, prescriptions ordered, laboratory or other ordered procedures, and

clinical narrative notes. Common data models can contain fully identified or de-identified data. In the latter case, patient identifiers have been stripped out and other alterations have been applied that will obfuscate the data and prevent the re-identification of patients (see Ethical, Legal, and Social Implications of Translational Medicine for a discussion of protecting patient privacy). For research use, data are typically (but not always) de-identified. The most commonly used data models include the Observational Health Data Sciences and Informatics (OHDSI) Observational Medical Outcomes Partnership (OMOP) data model (Gini et al. 2016), the Patient-Centered Outcomes Research Institute data model (Fleurence et al. 2014), and schema based on Fast Healthcare Interoperability Resources, or FHIR (Mandel et al. 2016); FHIR is discussed in more detail below. As of this writing, more than 650 million patient medical records are in the OMOP format. OMOP can be used to identify groups of patients fitting certain criteria, such as age, gender, past diagnoses, medications ordered, and other clinical criteria. The OHDSI project provides a number of software packages that can be used for accessing the OMOP data model data, including the Atlas data browser and the Achilles visual data quality viewer.

Much of Electronic Health Record Data are Plaintext

Medical records contain a wealth of text data in addition to structured (or coded) data. These textual data can be a note that describes a patient encounter, the patient's family history, any known drug allergies, their medical history, pathology reports, and many other pieces of clinically relevant information. Performing research analyses based on clinical notes can be difficult for several reasons. First, clinical notes can be identifying – that is, they may contain patient identifiers like name and age information that researchers may not be legally or ethically allowed to access. Second, clinical notes, being text, are difficult to analyze even when using sophisticated natural language processing (NLP) algorithms. Third, clinical notes are of uncertain research significance and their utility may be unclear. All this aside, notes are quite important for translational research purposes, as much of the patient's phenotypic descriptions are embedded there. For example, terms such as “diarrhea,” “vomiting,” or “fever” may only be found within note text. Tools such as MetaMap and others (Chiaramello et al. 2016) enable recognition of concepts from ontologies not generally coded in health records. One such ontology is the HPO, which contains concepts describing phenotypes including general terms like vomiting or diarrhea and very specific terms like “Abnormal serum insulin-like growth factor 1 level.”

Informatics and Precision Medicine

Describing Patient Phenotype

Ironically, EHR data are not necessarily collected to accurately describe patient phenotype, meaning they are an under-used (or altogether missed) opportunity to advance human disease research (Jensen et al. 2012). The reasons for this are complex. First, health records are often used for billing, managing and tracking orders, and recording events that occur during patient encounters. However, the collection of phenotypic data could span many encounters, and ongoing problems may not be covered in a specific encounter; for example, a patient with lupus might have been treated at an outpatient clinic for a seasonal cold, but no mention of lupus is made in the notes for that visit. Second, the status of a phenotype can be very difficult to determine even if a record of it exists. For example, a diagnosis billing code for a form of cancer indicates something to do with cancer but does not specify whether the patient was tested for cancer, had cancer and is in remission, has ongoing cancer, or is being diagnosed with cancer. Further, exposures such as tobacco use, alcohol use, and other behaviors may not be encoded into the record at all. Box 17.2 provides an example of how well-phenotyped patients can be integrated with genetics to discover new associations.

Box 17.2 Associating Clinical Phenotypes to Variants: The PheWAS Approach

Biobanks have become increasingly popular as part of precision medicine programs at academic medical centers. These resources generally contain a population of consented or de-identified patients (who have given their permission to use their data and samples they have provided for general research use), blood, and clinical health record data. If many of the patients are genotyped or sequenced, association between traits (defined as phenotypes derived from health records) and variants in the population can be found. This was done at Vanderbilt University with the creation of BioVU, a DNA biobank. Here, genotyping of thousands of patients in BioVU has been used as a first step toward association of variants to ICD diagnosis codes in their medical records. From this, seven phenotypes were replicated from previous genome-wide association studies (GWAS) (Denny et al. 2010): atrial fibrillation, Crohn disease, carotid artery stenosis, coronary artery disease, multiple sclerosis, systemic lupus erythematosus, and rheumatoid arthritis. Since then, many other studies have been performed to discover new relationships that have the potential to improve health outcomes.

Drug Repurposing

Informatics approaches enable the development of methods for hypothesizing the potential new uses for previously approved drugs based on shared mechanisms (Dudley et al. 2011). For example, thalidomide has been approved as a sedative and to treat leprosy (Laffitte and Revuz 2004); it was then repurposed and approved to treat multiple myeloma in 2012. If two conditions share a common mechanism (such as aberrant phosphorylation by a specific kinase), then therapies that target one condition may also apply to the other. This approach has been applied to genomic, proteomic, and other high-throughput molecular datasets of diseases and disease models, yielding databases of shared pathways and other mechanisms that can be used to find potential new uses for existing drugs. Similarly, an alternative approach is to use small molecules or drugs as perturbagens, interrupting intercellular processes in order to measure molecular changes that they induce when administered to a specific cell line or organism; these changes could include alterations in transcriptome gene expression. These high-throughput screens can provide insights into similar mechanisms between disease and a potential therapeutic. The Drug Repurposing Hub hosted at the Broad Institute is one such resource for accessing large amounts of screening data that quantify the molecular impacts of drugs and can be used to compare with patient-specific or other transcriptomic datasets (Corsello et al. 2017). As of this writing, the Drug Repurposing Hub had 6125 experimentally verified compounds available for analysis.

Clinical Marker Development from -omics Data

Genomic, proteomic, and metabolomic technologies have enabled the development of molecular marker technologies that can identify undiagnosed conditions or treatment paths for patients. These markers could include genetic variants, expressed transcripts, proteins or fragments of proteins, metabolites, or even specific microorganisms from the human microbiota (Chapter 16). For example, the Clinical Proteomic Tumor Analysis Consortium (CPTAC) (Edwards et al. 2015), an NIH-funded consortium, is developing both the technologies and the data to find proteomic biomarkers in cancer for early detection, diagnosis, and treatment quickly and inexpensively. Markers of disease can be developed from any biochemically measured feature and can be associated to a diagnosis or treatment outcome. Box 17.3 provides an example, focusing on the descriptions of markers of biological aging.

Box 17.3 The Markers of Aging

Investigators have spent their careers trying to understand the basic mechanisms of human and model organism aging. Finding markers of human aging is still considered a grand challenge of both biology and medicine. High-throughput studies have used transcriptomes, proteomes, genetics, epigenetics, and metabolomics to find risk factors for aging.

Marker 1: telomeres. Telomeric repeats form the end of chromosomes and their length reduces with cell division. The telomerase protein extends these repeats, while shortening or completely losing the repeats is associated with apoptosis or senescence. Shortened telomeres are associated with aging and enriched shortening (such as with a dysfunctional mutant telomerase) is associated with aging phenotypes (Aubert and Lansdorp 2008).

Marker 2: DNA methylation. More recently, high-throughput measurements of epigenetic DNA methylation sites (5-methylcytosine) have been found to be markers of aging. In aging fibroblasts, DNA methylation has been found to decrease with age, while immortal cells maintain a steady level of methylation (Wilson and Jones 1983). This work was later supported by a study on an aging cohort of elderly subjects between the ages of 55 and 92 years of age (Bollati et al. 2009).

Integration of Heterogeneous Data Sources

Recently, there has been increasing interest in integrating multiple datasets together that can improve descriptions of patient phenotypes (Murdoch and Detsky 2013). These include behavioral datasets, such as active or passive mobile health datasets, social media data, genetic data, environmental exposure data, patient-reported outcome measures (PROMs) generated through the use of survey instruments, or other patient-reported data (such as images provided by the patient). As a relevant example, wearable sensors can be used to provide “early warnings” to patients with bipolar disorder (Prociow et al. 2012). In addition, there have been a number of recent successes arising from the integration of datasets, such as prediction of behavior using cellphone usage (Prociow et al. 2012), identification of adverse drug events from social media data (Nikfarjam et al. 2015), or prediction of cardiac arrest incidence from genetic variant and neighborhood conditions (Mooney et al. 2016a). It is expected that the integration of medically relevant datasets will continue to improve our power to detect health risks and to better describe outcomes through a better description of patient phenotype and a patient’s environment.

Precision Medicine Initiatives

Published clinical studies typically compute average effects observed across a population of humans. These effects differ from the effect that would be seen in randomly selected individuals from that population or in a single patient in a clinic. There are likely to be ethnic and other genetic differences, differences in socioeconomic status, lifestyle, risk profile, and others (e.g. patient diversity in chronic kidney disease; Norris and Nissenson 2008). Genetic similarity, for example, clusters by place of origin or ancestry, and clusters poorly with race (Jorde and Wooding 2004). All patients are different, and we are just beginning to learn how to treat a patient individually tailored to their person (the “*N* of 1” problem). As may have become quite obvious by this point, translation of the population-based body of clinical knowledge to individual patients in a data-driven way is difficult (Hamburg and Collins 2010). In order to better assess this translational potential, large cohorts of volunteer participants are being recruited to studies to assess general population-based risk factors, among other goals. One such example is the NIH’s All of Us project, currently under way in the United States with the goal of enrolling 1 million volunteers who agree to provide their clinical, genetic,

mobile health data (mHealth), PROMs, biometric, and other relevant clinical data (Ashley 2015). The All of Us initiative is likely to provide important new insights into risk factors for disease and deeper insights into outcomes of interventions developed using clinical studies. All of Us is positioning itself as a model for open science and data sharing, enabling data scientists access to richer data than have been widely available previously for any study and, in the process, bringing a new set of challenges to the fore (Adams and Petersen 2016). It is expected that genome sequences will be a core component of this research platform. Other initiatives include P4 Medicine at the Institute of Systems Biology (Flores et al. 2013) and the development of biobanks such as MyCode by the Geisinger Health System (Carey et al. 2016) and BioVu at Vanderbilt University (Cronin et al. 2014).

Community Challenges Solve Innovative Problems Collaboratively

One approach toward solving challenging problems in translational informatics is to engage the community of “citizen scientists” to bring forward unique and novel solutions (Saez-Rodriguez et al. 2016). An ecosystem of data challenges has emerged across the informatics domain. These challenges are often public and open to all participants. This community first emerged from the Critical Assessment of Structure Prediction (CASP) (Moult et al. 2011), a biannual unbiased assessment of the best methods for predicting macromolecular structures. Since then, a number of other “critical assessments” have emerged, as well as from other organizations, such as the DREAM challenges (Jarchum and Jones 2015), and even for-profit companies focusing on hosting challenges (e.g. Kaggle). Challenges are an excellent way to stimulate new methodology and innovation, to build collaborations, and to assess methodology in an unbiased manner. Now there are challenges that range from the development of new bioinformatic analysis tools, methods for clinical decision support, and methods for NLP and genomics, among others. One notable family of challenges is the regularly occurring Critical Assessment of Genome Interpretation (CAGI), an effort to assess tools for inferring phenotype from human genetic sequences. Box 17.4 presents an example of the CAGI whole genome prediction challenge using data derived from personal genomes.

Box 17.4 The CAGI Personal Genome Project Community Challenge

The Personal Genome Project (PGP) (Ball et al. 2014) was developed to enable research on human genomes and encourages participation from individuals who adopt “open consent.” Open consent allows personal genomes, traits, and other information to be released for general research use. The PGP enrolls participants, has them sign on to open consent, has them fill out a detailed health survey, takes a DNA sample, and then sequences and publishes their genome. The Critical Assessment of Genome Interpretation (CAGI) organizers saw this as an opportunity to assess how well consumer genetic testing and other methods work at predicting traits directly from a genome sequence, and the PGP challenge was born. In 2010, the challenge was simple: the first 10 PGP genomes were released without the health survey data to challenge method developers. Developers then attempted to predict both binary and numerical traits from the sequences. Once the predictions were collected, the health surveys were made public and the accuracy of the predictions was assessed. Subsequently, between 2010 and 2016, two other PGP challenges were held where developers were asked to simply match a known trait profile (again derived from health survey) to a genome. The list of profiles contained many decoys, making matching even more difficult. A number of teams from across the globe submitted predictions. Over the three challenges, many lessons were learned (Figure 17.2). First, matching an extensive profile for an individual to a genome is a challenging problem and the best methods were only able to match approximately 20% of the genome profile pairs. Second, individual traits are exceedingly difficult to predict, and sometimes just knowing how common a trait is in a population is a challenge. Still, accuracy increased over time for the challenges.

Electronic Health Record Systems can be Customized

One of the goals of translational medicine is the translation or implementation (use) of new research findings back to the clinic. Traditionally, electronic medical record systems have been difficult to engineer and customize for a specific clinical site. EHR systems have mechanisms for extension and decision support built in, such as patient care provider alerting based on rules. Alerts can be customized to occur when certain defined events occur within the system and rules that trigger alerts can be customized to support notification of potential medication interactions, reminders for laboratory tests or procedures, pharmacogenetics indications, and others (Nishimura et al. 2015). Pharmacogenetic alerts, for example, can facilitate personalized drug dosing using genetic data, and applications of these alerts were seen as advantageous by prescribing physicians (Overby et al. 2015). Alerting with too much frequency in EHR systems causes “alert fatigue,” where healthcare providers begin to ignore alerts as they become more common, particularly when providing alerts for obvious or false-positive actions. More recently, EHR systems have become even more customizable and can be standardized using standard application programming interfaces (APIs) that are interoperable across different system vendors. The most widely used standard is the Fast Healthcare Interoperability Resources (FHIR, pronounced “fire”) (Mandel et al. 2016), a communication and data standard being developed for extending EHRs. FHIR is enhanced by Substitutable Medical Apps and Reusable Technologies (SMARTs) that enable building novel applications that use FHIR APIs (such as mobile applications) (Bloomfield et al. 2017). While “SMART On FHIR” is still in development, it is based on the HL7 communication

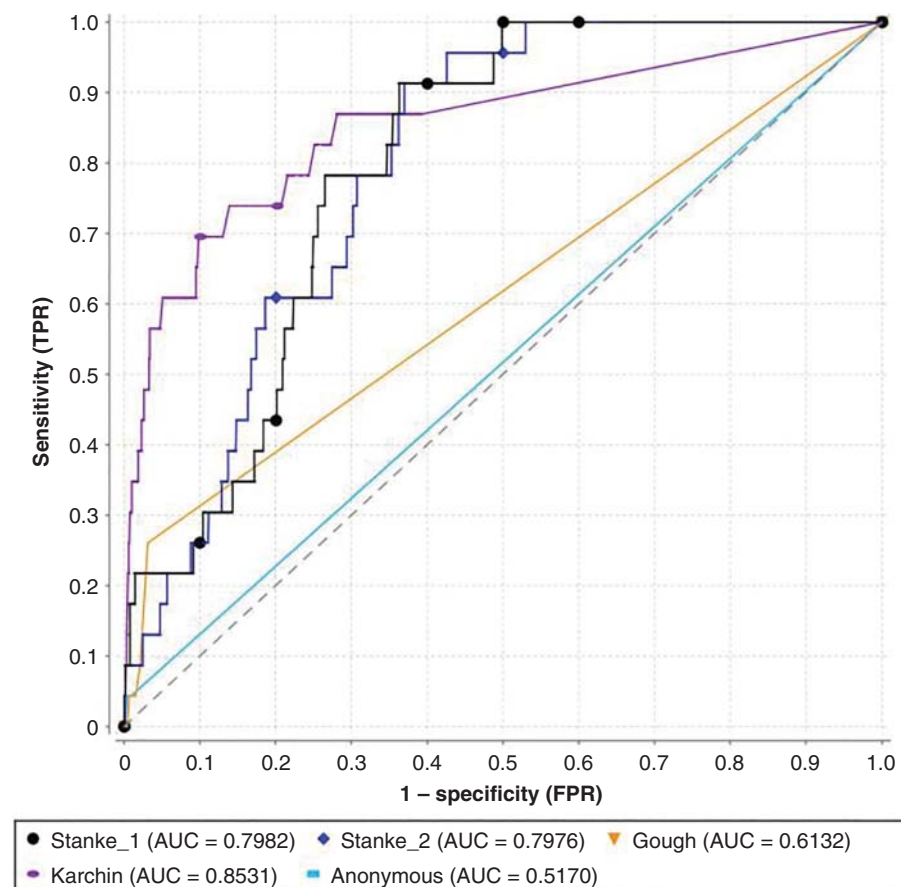


Figure 17.2 Receiver operating characteristic (ROCs) curves of five submissions from the CAGI PGP 2015 matching challenge. Challenge developers submitted genome health profile match probabilities, and these were used to compare against the actual known genotype–phenotype pairs. AUC, area under the curve. See Cai et al. (2017) for more detail.

standard for communicating clinical data over the internet and is becoming widely supported. SMART On FHIR represents a giant leap forward for translational informatics, as it opens the interoperability of extensions of health record systems considerably.

Informatics for Prevention Policy

While much of the effort in the field of translational informatics has been focused on clinical applications, biomedical informatics also has the potential to inform policy, particularly with respect to prevention. For example, EHRs were used to efficiently study the collision risk elevation connoted by specific medications (Rudisill et al. 2016). More broadly, deploying novel information-gathering devices (e.g. air quality monitors mounted on public transportation vehicles to provide real-time, spatially detailed estimates of air pollution in an urbanized area; Devarakonda et al. 2013) and reusing information not initially gathered for health purposes (Hipp et al. 2013; Lovasi et al. 2013; Mooney et al. 2016b) – so-called “effluent data” (Mooney and Pejaver 2017) – is a mainstay of modern public health informatics research (Eysenbach 2009; Lazer et al. 2014; Santillana et al. 2014).

Ethical, Legal, and Social Implications of Translational Medicine

Given that the methods and results of translational bioinformatic research can have a direct impact on patient care, there is now a much greater interest in the ethical, legal, and social implications (ELSI) issues arising from the work being done in this field. These issues have been heavily discussed in the context of human genome sequencing (Collins 1999) but are equally important in the context of translational medicine. This chapter is not meant to give the reader a complete background in ELSI but, instead, to highlight that special care must be taken when developing directly translatable methods to patient care when using genomics data. We direct the reader to several of the many good reviews available discussing the impact of ELSI in genomics (Oliver and McGuire 2011; Callier et al. 2016) for a more complete treatment of the subject.

There are a number of questions that should be considered when releasing new methods, even if the method is intended for research use only. This includes what effect any new method will have on patients or providers, or whether there are any unintended consequences of such a tool. If the method is misused clinically, could patient harm come from its use? For example, what are the risks of false positives? Finally, there are potential regulatory issues that can vary from country to country that regulate the deployment of new clinical methods or return of clinical results to patients.

Protecting Patient Privacy

One of the primary difficulties in performing research on patient-derived data is the protection of the privacy of patients, research participants, and their family members. When research is performed, patient identifiers are generally removed; these identifiers include the patient’s name, U.S. Social Security number, medical record number, address, phone number, and similar personal data. In the United States, the Health Insurance Portability and Accountability Act (HIPAA) of 1996 defines 18 identifiers that can be stripped from a dataset to remove direct patient identification, and such datasets are called “limited.” De-identified datasets can be created from limited datasets by further anonymization, including randomly shifting dates of service or other transformations that obfuscate the data and make them difficult to directly re-identify the patient from whom the data were originally derived. (Note that institutional policies, human subjects research compliance regulations, and specific laws can vary on approaches to de-identification.) It is every researcher’s responsibility to ensure that patient confidentiality is kept; this includes ensuring information security, not

attempting to re-identify patients, and using “honest broker” approaches when accessing data. (Honest brokers are independent personnel who only provide data for research that have been approved by an Institutional Review Board.)

Summary

The future of translational bioinformatics is bright and will only continue to grow. As we learn more about the causes of disease through high-throughput experimentation, we begin to develop new methods for early detection and diagnosis, new interventions, and new tools for returning results to patients. We have become very good at sequencing and interpreting genomes clinically. This has led to the development of many tools for prediction and prioritization of pathogenic or functional variants and disease-associated genes. We have begun to connect genotype to phenotype better by connecting our rich knowledge of genetics to new sophisticated models of patient phenotype using EHRs, behavioral data, and patient-reported data, among others. This work has led to an explosion of activities surrounding computing with patient data. Some emergent projects have included drug-repurposing efforts, community challenges and open science, and useful clinical marker discovery. Further, using new technologies in EHR systems, such as standard APIs, we are able to implement these findings as data analysis and recommendation methods to directly impact clinical care by presenting this information to either the patient or the provider through the health record or applications developed ancillary to the health record.

Internet Resources

Kaggle	A for-profit website for hosting community challenges	Kaggle.com
Promethease	A method for predicting phenotypes from genetic data	www.snpedia.com/index.php/Promethease

References

- Adams, S.A. and Petersen, C. (2016). Precision medicine: opportunities, possibilities, and challenges for patients and providers. *J. Am. Med. Inf. Assoc.* 23 (4): 787–790.
- Adzhubei, I., Jordan, D.M., and Sunyaev, S.R. (2013). Predicting functional effect of human missense mutations using PolyPhen-2. *Curr. Protoc. Hum. Genet.* 76: 7.20.1–7.20.41.
- Anonymous (1996). Revisions of the International classification of diseases (ICD-9 and ICD-10): impact on health statistics. *Epidemiol. Bull.* 17 (2): 1–5.
- Ashley, E.A. (2015). The precision medicine initiative: a new national effort. *JAMA* 313 (21): 2119–2120.
- Aubert, G. and Lansdorp, P.M. (2008). Telomeres and aging. *Physiol. Rev.* 88 (2): 557–579.
- Ball, M.P., Bobe, J.R., Chou, M.F. et al. (2014). Harvard personal genome project: lessons from participatory public research. *Genome Med.* 6 (2): 10.
- Bloomfield, R.A. Jr., Polo-Wood, F., Mandel, J.C., and Mandl, K.D. (2017). Opening the Duke electronic health record to apps: implementing SMART on FHIR. *Int. J. Med. Inf.* 99: 1–10.
- Bollati, V., Schwartz, J., Wright, R. et al. (2009). Decline in genomic DNA methylation through aging in a cohort of elderly subjects. *Mech. Ageing Dev.* 130 (4): 234–239.
- Boycott, K.M., Vanstone, M.R., Bulman, D.E., and MacKenzie, A.E. (2013). Rare-disease genetics in the era of next-generation sequencing: discovery to translation. *Nat. Rev. Genet.* 14 (10): 681–691.

- Bromberg, Y., Yachdav, G., and Rost, B. (2008). SNAP predicts effect of mutations on protein function. *Bioinformatics* 24 (20): 2397–2398.
- Cai, B., Li, B., Kiga, N. et al. (2017). Matching phenotypes to whole genomes: lessons learned from four iterations of the personal genome project community challenges. *Hum. Mutat.* 38 (9): 1266–1276.
- Callier, S.L., Abudu, R., Mehlman, M.J. et al. (2016). Ethical, legal, and social implications of personalized genomic medicine research: current literature and suggestions for the future. *Bioethics* 30 (9): 698–705.
- Carey, D.J., Fetterolf, S.N., Davis, F.D. et al. (2016). The Geisinger MyCode community health initiative: an electronic health record-linked biobank for precision medicine research. *Genet. Med.* 18 (9): 906–913.
- Cariaso, M. and Lennon, G. (2012). SNPedia: a wiki supporting personal genome annotation, interpretation and analysis. *Nucleic Acids Res.* 40 (Database issue): D1308–D1312.
- Chiaramello, E., Pincioli, F., Bonalumi, A. et al. (2016). Use of “off-the-shelf” information extraction algorithms in clinical informatics: a feasibility study of MetaMap annotation of Italian medical notes. *J. Biomed. Inf.* 63: 22–32.
- Collins, F.S. (1999). Shattuck Lecture – medical and societal consequences of the Human Genome Project. *N. Engl. J. Med.* 341 (1): 28–37.
- Collins, A., Lonjou, C., and Morton, N.E. (1999). Genetic epidemiology of single-nucleotide polymorphisms. *Proc. Natl. Acad. Sci. U.S.A.* 96 (26): 15173–15177.
- Cooper, G.M. and Shendure, J. (2011). Needles in stacks of needles: finding disease-causal variants in a wealth of genomic data. *Nat. Rev. Genet.* 12 (9): 628–640.
- Corsello, S.M., Bittker, J.A., Liu, Z. et al. (2017). The drug repurposing hub: a next-generation drug library and information resource. *Nat. Med.* 23 (4): 405–408.
- Cronin, R.M., Field, J.R., Bradford, Y. et al. (2014). Phenome-wide association studies demonstrating pleiotropy of genetic variants within FTO with and without adjustment for body mass index. *Front. Genet.* 5: 250.
- Denny, J.C., Ritchie, M.D., Basford, M.A. et al. (2010). PheWAS: demonstrating the feasibility of a phenome-wide scan to discover gene-disease associations. *Bioinformatics* 26 (9): 1205–1210.
- Devarakonda, S., Sevusu, P., Liu, H. et al. (2013). Real-time air quality monitoring through mobile sensing in metropolitan areas. In: *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing, Chicago, IL (11 August 2013)*. New York, NY: ACM.
- Dudley, J.T., Deshpande, T., and Butte, A.J. (2011). Exploiting drug-disease relationships for computational drug repositioning. *Briefings Bioinf.* 12 (4): 303–311.
- Edwards, N.J., Oberti, M., Thangudu, R.R. et al. (2015). The CPTAC data portal: a resource for cancer proteomics research. *J. Proteome Res.* 14 (6): 2707–2713.
- Eysenbach, G. (2009). Infodemiology and infoveillance: framework for an emerging set of public health informatics methods to analyze search, communication and publication behavior on the Internet. *J. Med. Internet Res.* 11 (1): e11.
- Fleurence, R.L., Curtis, L.H., Califf, R.M. et al. (2014). Launching PCORnet, a national patient-centered clinical research network. *J. Am. Med. Assoc.* 311 (4): 578–582.
- Flores, M., Glusman, G., Brogaard, K. et al. (2013). P4 medicine: how systems medicine will transform the healthcare sector and society. *Per. Med.* 10 (6): 565–576.
- Forbes, S.A., Beare, D., Boutselakis, H. et al. (2017). COSMIC: somatic cancer genetics at high-resolution. *Nucleic Acids Res.* 45 (Database issue): D777–D783.
- Gini, R., Schuemie, M., Brown, J. et al. (2016). Data extraction and management in networks of observational health care databases for scientific research: a comparison of EU-ADR, OMOP, mini-sentinel and MATRICE strategies. *EGEMS (Wash DC)* 4 (1): 1189.
- Hamburg, M.A. and Collins, F.S. (2010). The path to personalized medicine. *N. Engl. J. Med.* 363 (4): 301–304.
- Hipp, J.A., Adlakha, D., Eyler, A.A. et al. (2013). Emerging technologies: webcams and crowd-sourcing to identify active transportation. *Am. J. Prev. Med.* 44 (1): 96.

- Ioannidis, N.M., Rothstein, J.H., Pejaver, V. et al. (2016). REVEL: an ensemble method for predicting the pathogenicity of rare missense variants. *Am. J. Hum. Genet.* 99 (4): 877–885.
- Jarchum, I. and Jones, S. (2015). DREAMing of benchmarks. *Nat. Biotechnol.* 33 (1): 49–50.
- Jensen, P.B., Jensen, L.J., and Brunak, S. (2012). Mining electronic health records: towards better research applications and clinical care. *Nat. Rev. Genet.* 13 (6): 395–405.
- Jha, A.K., DesRoches, C.M., Campbell, E.G. et al. (2009). Use of electronic health records in U.S. hospitals. *N. Engl. J. Med.* 360 (16): 1628–1638.
- Jorde, L.B. and Wooding, S.P. (2004). Genetic variation, classification and “race”. *Nat. Genet.* 36 (11 Suppl): S28–S33.
- Kohler, S., Vasilevsky, N.A., Engelstad, M. et al. (2017). The human phenotype ontology in 2017. *Nucleic Acids Res.* 45 (D1): D865–D876.
- Laffitte, E. and Revuz, J. (2004). Thalidomide: an old drug with new clinical applications. *Expert Opin. Drug Saf.* 3 (1): 47–56.
- Landrum, M.J., Lee, J.M., Benson, M. et al. (2016). ClinVar: public archive of interpretations of clinically relevant variants. *Nucleic Acids Res.* 44 (D1): D862–D868.
- Lazer, D., Kennedy, R., King, G., and Vespignani, A. (2014). The parable of Google Flu: traps in big data analysis. *Science* 343 (6176): 1203–1205.
- Li, B., Krishnan, V.G., Mort, M.E. et al. (2009). Automated inference of molecular mechanisms of disease from amino acid substitutions. *Bioinformatics* 25 (21): 2744–2750.
- Liu, X., Wu, C., Li, C., and Boerwinkle, E. (2016). dbNSFP v3.0: a one-stop database of functional predictions and annotations for human nonsynonymous and splice-site SNVs. *Hum. Mutat.* 37 (3): 235–241.
- Lovasi, G.S., O’Neil-Dunne, J.P., Lu, J.W. et al. (2013). Urban tree canopy and asthma, wheeze, rhinitis, and allergic sensitization to tree pollen in a New York City birth cohort. *Environ. Health Perspect.* 121 (4): 494.
- Lugo-Martinez, J., Pejaver, V., Pagel, K.A. et al. (2016). The loss and gain of functional amino acid residues is a common mechanism causing human inherited disease. *PLoS Comput. Biol.* 12 (8): e1005091.
- Mandel, J.C., Kreda, D.A., Mandl, K.D. et al. (2016). SMART on FHIR: a standards-based, interoperable apps platform for electronic health records. *J. Am. Med. Inf. Assoc.* 23 (5): 899–908.
- Mooney, S.D. and Klein, T.E. (2002). The functional importance of disease-associated mutation. *BMC Bioinf.* 3: 24.
- Mooney, S.J. and Pejaver, V. (2017). Big data in public health: terminology, machine learning, and privacy. *Annu. Rev. Public Health* 39: 95–112.
- Mooney, S.J., Grady, S.T., Sotoodehnia, N. et al. (2016a). In the wrong place with the wrong SNP: the association between stressful neighborhoods and cardiac arrest within Beta-2-adrenergic receptor variants. *Epidemiology* 27 (5): 656–662.
- Mooney, S.J., DiMaggio, C.J., Lovasi, G.S. et al. (2016b). Use of Google Street View to assess environmental contributions to pedestrian injury. *Am. J. Public Health* 106 (3): 462–469.
- Mort, M., Sterne-Weiler, T., Li, B. et al. (2014). MutPred splice: machine learning-based prediction of exonic variants that disrupt splicing. *Genome Biol.* 15 (1): R19.
- Moult, J., Fidelis, K., Kryshtafovych, A., and Tramontano, A. (2011). Critical assessment of methods of protein structure prediction (CASP)--round IX. *Proteins* 79 (Suppl 10): 1–5.
- Mungall, C.J., McMurry, J.A., Köhler, S. et al. (2017). The Monarch initiative: an integrative data and analytic platform connecting phenotypes to genotypes across species. *Nucleic Acids Res.* 45 (D1): D712–D722.
- Murdoch, T.B. and Detsky, A.S. (2013). The inevitable application of big data to health care. *JAMA* 309 (13): 1351–1352.
- Musen, M.A., Noy, N.F., Shah, N.H. et al. (2012). The National Center for Biomedical Ontology. *J. Am. Med. Inf. Assoc.* 19 (2): 190–195.
- Ng, P.C. and Henikoff, S. (2003). SIFT: predicting amino acid changes that affect protein function. *Nucleic Acids Res.* 31 (13): 3812–3814.

- Nikfarjam, A., Sarker, A., O'Connor, K. et al. (2015). Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. *J. Am. Med. Inf. Assoc.* 22 (3): 671–681.
- Nishimura, A.A., Shirts, B.H., Dorschner, M.O. et al. (2015). Development of clinical decision support alerts for pharmacogenomic incidental findings from exome sequencing. *Genet. Med.* 17 (11): 939–942.
- Norris, K. and Nissenson, A.R. (2008). Race, gender, and socioeconomic disparities in CKD in the United States. *J. Am. Soc. Nephrol.* 19 (7): 1261–1270.
- Oliver, J.M. and McGuire, A.L. (2011). Exploring the ELSI universe: critical issues in the evolution of human genomic research. *Genome Med.* 3 (6): 38.
- Overby, C.L., Devine, E.B., Abernethy, N. et al. (2015). Making pharmacogenomic-based prescribing alerts more effective: a scenario-based pilot study with physicians. *J. Biomed. Inf.* 55: 249–259.
- Pathak, J., Kho, A.N., and Denny, J.C. (2013). Electronic health records-driven phenotyping: challenges, recent advances, and perspectives. *J. Am. Med. Inf. Assoc.* 20 (e2): e206–e211.
- Prociow, P., Wac, K., and Crowe, J. (2012). Mobile psychiatry: towards improving the care for bipolar disorder. *Int. J. Ment. Health Syst.* 6 (1): 5.
- Radivojac, P., Peng, K., Clark, W.T. et al. (2008). An integrated approach to inferring gene-disease associations in humans. *Proteins* 72 (3): 1030–1037.
- Radivojac, P., Clark, W.T., Oron, T.R. et al. (2013). A large-scale evaluation of computational protein function prediction. *Nat. Methods* 10 (3): 221–227.
- Relling, M.V. and Klein, T.E. (2011). CPIC: clinical pharmacogenetics implementation consortium of the pharmacogenomics research network. *Clin. Pharmacol. Ther.* 89 (3): 464–467.
- Richards, S., Aziz, N., Bale, S. et al. (2015). Standards and guidelines for the interpretation of sequence variants: a joint consensus recommendation of the American College of Medical Genetics and Genomics and the Association for Molecular Pathology. *Genet. Med.* 17 (5): 405–424.
- Ritchie, G.R.S., Dunham, I., Zeggini, E., and Flicek, P. (2014). Functional annotation of noncoding sequence variants. *Nat. Methods* 11 (3): 294–296.
- Rudisill, T.M., Zhu, M., Davidov, D. et al. (2016). Medication use and the risk of motor vehicle collision in West Virginia drivers 65 years of age and older: a case-crossover study. *BMC Res. Notes* 9: 166.
- Saez-Rodriguez, J., Costello, J.C., Friend, S.H. et al. (2016). Crowdsourcing biomedical research: leveraging communities as innovation engines. *Nat. Rev. Genet.* 17 (8): 470–486.
- Santillana, M., Zhang, D.W., Althouse, B.M., and Ayers, J.W. (2014). What can digital disease detection learn from (an external revision to) Google Flu Trends? *Am. J. Prev. Med.* 47 (3): 341–347.
- Smith, B., Ashburner, M., Rosse, C. et al. (2007). The OBO foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat. Biotechnol.* 25 (11): 1251–1255.
- Stenson, P.D., Mort, M., Ball, E.V. et al. (2017). The Human Gene Mutation Database: towards a comprehensive repository of inherited mutation data for medical research, genetic diagnosis and next-generation sequencing studies. *Hum. Genet.* 136 (6): 665–677.
- Thorn, C.F., Klein, T.E., and Altman, R.B. (2010). Pharmacogenomics and bioinformatics: PharmGKB. *Pharmacogenomics* 11 (4): 501–505.
- Tranchevent, L.C., Ardeshirdavani, A., ElShal, S. et al. (2016). Candidate gene prioritization with Endeavour. *Nucleic Acids Res.* 44 (W1): W117–W121.
- Vuokko, R., Mäkelä-Bengs, P., Hyppönen, H., and Doupi, P. (2015). Secondary use of structured patient data: interim results of a systematic review. *Stud. Health Technol. Inf.* 210: 291–295.
- Wang, Z. and Moulton, J. (2003). Three-dimensional structural location and molecular functional effects of missense SNPs in the T cell receptor Vbeta domain. *Proteins* 53 (3): 748–757.
- Wang, X., Wang, G., Shen, C. et al. (2008). Using RNase sequence specificity to refine the identification of RNA-protein binding regions. *BMC Genomics* 9 (Suppl 1): S17.

- Wang, K., Li, M., and Hakonarson, H. (2010). ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Res.* 38 (16): e164.
- Warde-Farley, D., Donaldson, S.L., Comes, O. et al. (2010). The GeneMANIA prediction server: biological network integration for gene prioritization and predicting gene function. *Nucleic Acids Res.* 38 (Web Server issue): W214–W220.
- Wilson, V.L. and Jones, P.A. (1983). DNA methylation decreases in aging but not in immortal cells. *Science* 220 (4601): 1055–1057.
- Wishart, D.S., Feunang, Y.D., Guo, A.C. et al. (2018). DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Res.* 46 (D1): D1074–D1082.
- Xin, J., Mark, A., Afrasiabi, C. et al. (2016). High-performance web services for querying gene and variant annotation. *Genome Biol.* 17 (1): 91.

18

Statistical Methods for Biologists

Hunter N.B. Moseley

Introduction

The dramatic growth in the generation and accumulation of biological and biomedical data during the twentieth century has created a fundamentally different data- and knowledge-rich research environment in the twenty-first century. To effectively function within this environment, biologists must be able to utilize large amounts of data and accumulated knowledge in their day-to-day research. These datasets often include thousands to millions to even billions of individual pieces of data that are simply too large for manual analysis. Therefore, it is critical for biologists to understand derived summative representations of these large datasets. A statistic such as the mean (or average) is a commonly used derived representation of a set of data, and the field of statistics is the science involving the derivation and application of useful statistics from datasets. In the context of bioinformatics, large datasets derived from a variety of “-omics” technologies or aggregated information in knowledge bases must be summarized by descriptive, summative representations that facilitate the evaluation of the dataset and its utilization in other analyses. Often times, these other analyses produce new information and knowledge from the dataset. But the fundamental understanding of a dataset, the experiments that produced the dataset, and the methods used to analyze the dataset are required to create accurate interpretations that form new information and knowledge. Statistics provides a critical perspective and set of concepts for developing this fundamental understanding of a dataset and how it may be used effectively.

Descriptive Representations of Data

Data vs. Information vs. Knowledge

Data, information, and knowledge are related but separate and distinct concepts. These terms are often used interchangeably, leading to confusion in exactly what is being collected, provided, or analyzed. Specifically, *data* are an unorganized collection of simple facts and observations. This definition of data begs the question: “What are observations?” An *observation* is an acquired measure of a phenomenon. For a statistician, the phenomenon is a statistical experiment. In a scientific research context, an observation is the determination of the amount or degree of some property or characteristic of a specific physical entity or event. Thus, data are an unorganized collection of such measurements. *Information* is data that are organized, analyzed, and interpreted into a useful form, often for making decisions. Finally, *knowledge* is information, understanding, and skills derived from education and experience in a particular domain or area of study. As illustrated in Figure 18.1, individual observations are collected into data, which are interpreted into useful information, which is further distilled into new knowledge.

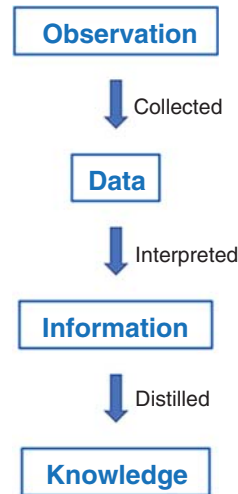


Figure 18.1 Relationships between observation, data, information, and knowledge. Source: Reproduced with permission of Hunter Moseley, <https://doi.org/10.6084/m9.figshare.4968125.v1>. Licensed under CC By 4.0.

From a data perspective, a collection of related observations representing the same or similar phenomenon is called a *random variable*. This term has a more abstract definition in statistics, where it is represented as a mathematical function that maps possible outcomes from a statistical experiment (phenomenon) to a measurable space of possible values (observations). Depending on what property or characteristic is being measured, random variables can be numerical or categorical, as shown in Figure 18.2. A numerical random variable is a range of possible measurable quantities. If the range is defined in terms of real numbers or a similar infinite number set, then the random numerical variable is continuous. In this context, continuity is a property of the mathematical function $f(x)$ representing the random variable, where the limit of $f(x)$ approaches $f(c)$ as x approaches c . For example, measuring the distance between donor and acceptor chromophores using fluorescence resonance energy transfer (FRET) would be a continuous random variable, since a continuous range of real numbers are possible observations (i.e. mapped outcomes of the statistical experiment representing a set of observations generated from the FRET analytical experiment). Another example of a continuous random variable is the length of time that a mouse searches for the platform in a Morris water maze test. Now if the range is defined in terms of integers (i.e. positive and/or negative counting numbers), then the random numerical variable is discrete. The number of cells counted in a flow cytometer is a discrete random variable within a discrete range of

Type	Random variable definition	Example
Numerical	Quantitatively measured observations	How many? How much?
Continuous	...within a range of real or complex values	[0.0, 10.0]; {1.50, 4.58, 9.45}
Discrete (cardinal)	...within a countable range of integer values	[0, 10]; {0, 3, 4, 7, 9, 10}
Categorical	Qualitative observations describing a characteristic or relative quality	What type? What category? What relative quality?
Ordinal	...with a logical order or ranking	{1 st , 2 nd , 3 rd }; {low, med, high}
Nominal	...without a logical sequence	{male, female}; {blue, green, red}

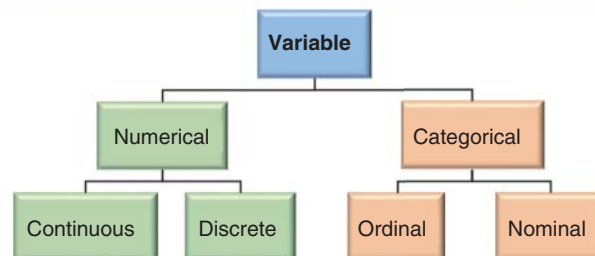


Figure 18.2 Types of variables and their hierarchical relationships. Source: Reproduced with permission of Hunter Moseley, <https://doi.org/10.6084/m9.figshare.4968143.v1>. Licensed under CC By 4.0.

natural numbers (i.e. non-negative integers). Another example is a “point count” or number of a type of bird seen or heard at a given observation station over a set period of time. A categorical random variable is a set of possible qualitatively measured observations describing a characteristic or relative quantity. If the values have a relative or logical order or ranking, then the categorical variable is considered ordinal. The relative order of nucleotides observed for a specific nucleotide sequence is an ordinal random variable. Another example is the reporting of the relative level of pain experienced by patients undergoing some procedure on a scale of 1 (most severe pain experienced) to 5 (no pain). If the values have no logical order, then the categorical variable is considered nominal. The sex of an animal is a nominal random variable that, in most cases, is limited to male and female. Another example is a human cell line with and without specific CRISPR-Cas9 gene knockouts.

Datasets and Data Schemas

A *dataset* and its older accepted spelling “data set” is simply a collection of related data and information. But the canonical definition of *dataset* refers to a collection of related sets of data and information that are organized with respect to observable phenomena (variables) and entities that relate observations across phenomena (i.e. multiple statistical experiments). This organization is usually represented in a two-dimensional matrix or relational table, where columns or fields represent distinct variables of data and rows represent distinct entities. For example, age, sex, race, weight, height, disease status, treatment, and other outcome variables can be collected for a set of properly consented human subjects and organized into a two-dimensional table where values for specific random variables (columns 2+) are associated with a specific human being (with de-identified subject ID in column 1) for use in a clinical trial (Figure 18.3a). In this context, the collection of inter-related observations from a single clinical, biomedical, biological, and/or analytical experiment are organized into these two-dimensional datasets. However, datasets can also refer to collections of canonical

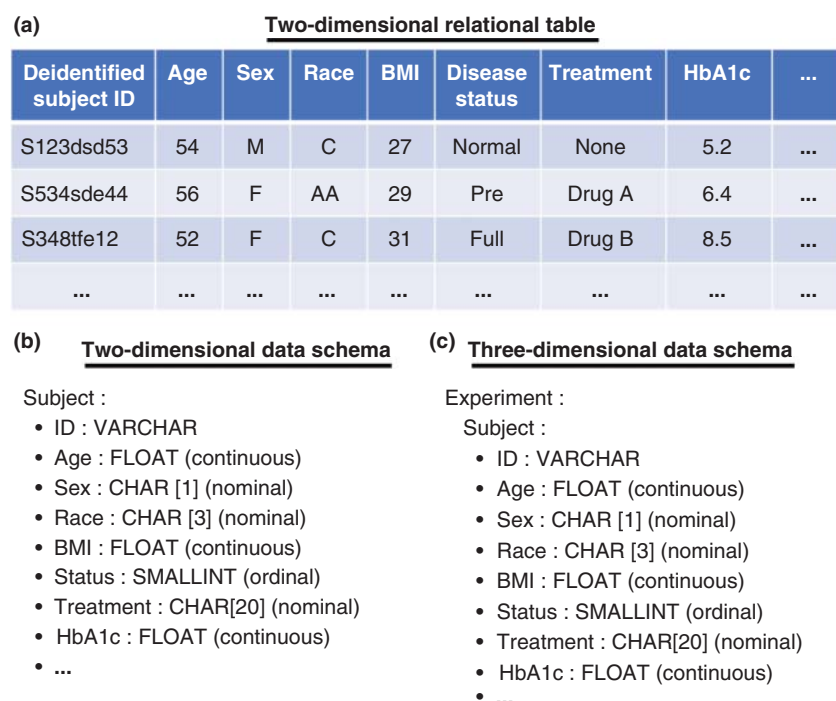


Figure 18.3 Organization of an example dataset. (a) Part of a two-dimensional (2D) relational table relating de-identified human subjects (rows) to specific sample variables (columns). (b) A 2D data schema showing the organization of the dataset and the types of variables. (c) A three-dimensional (3D) data schema showing an additional relationship (dimension) of subjects to biological/analytical experiment. Source: Reproduced with permission of Hunter Moseley, <https://doi.org/10.6084/m9.figshare.4968146.v1>. Licensed under CC By 4.0.

datasets, in which individual biological and/or analytical experiments represent a third dimension. Therefore, a description of the organization of the dataset, called a *data schema*, is required for analysis and interpretation of a dataset (Figure 18.3b,c). Also, a data schema is often referred to as a data dictionary, database schema, or metadata, depending on the context of use. As datasets continue to grow in size and complexity, the quality of the data organization and data schema often becomes a limiting factor in the usability of a dataset.

As previously mentioned, datasets are usually too large to examine and understand by manual inspection. Therefore, summative, descriptive representations of a dataset are required for their evaluation and interpretation. Three major types of descriptive data representation are data schemas, descriptive statistics, and graphs. As a starting point, a data schema provides a very good descriptive overview of a dataset. From a well-described data schema, the number and specific types of variables are easily determined. Also, the organization of variables with respect to entity – the subject – becomes evident (Figure 18.3b), and higher order organizations relating variables and/or subjects across biological and analytical experiments can then be deduced (Figure 18.3c).

Descriptive Statistics

A *descriptive statistic* is a single measurable characteristic that quantitatively describes or summarizes a collection of related data (Daniel and Wayne 1995). However, in a strict statistical definition, there are two related concepts: statistic and parameter. A statistic is a single measure of some sample variable or measurable sample attribute, where a sample is a subset of entities from a population. The term *parameter* is reserved for a characteristic or attribute of a population that often cannot be directly measured. Most datasets only contain data representing a subset of a population, known as a sample. For example, the mean height of 1000 female tennis players would represent the mean height statistic for the sample of 1000 tennis players, which could be used to infer the mean height parameter for the population of all female tennis players. But a dataset could also contain all data for a finite population like “all human employees of a given business.” In this case, a parameter for this finite population could be directly measured from the dataset and not just estimated from a sample statistic. However, such narrowly defined populations can also be viewed as a sample of a much larger population, like “all humans on the planet” or even “all humans who have ever lived or possibly could live.” Therefore, classifying a measurable descriptive characteristic of given dataset variable as a statistic versus a parameter is a matter of perspective.

Figure 18.4 provides a list of the most commonly used descriptive statistics for collections of data representing sample variables. The first descriptive statistic in the light blue row is the size (or cardinality) of a collection of data. The importance of size cannot be overstated, as it represents the most direct measure of the quantity of data present in a set of related data – the variable. The data quantity, in turn, typically limits the information content of the variable. The next most commonly used descriptive statistics, shown in the light green rows in Figure 18.4, are called statistics of central tendency. In statistics, central tendency is the typical, central, expected value for a collection of values or a range of possible values. The best known statistic of central tendency is the arithmetic mean or average for a collection of values. For example, the mean of {3.2, 4.1, 4.1, 4.2, 4.4, 5.1, 5.1, 5.4, 5.4, 5.5, 5.8, 5.8, 6.2, 7.0, 7.5} is 5.25, which represents a fairly typical value for this collection of values. In this context, the central or typical value represents the most frequently occurring value or set of values within a distribution, which is the set of frequencies for all possible values that may occur. This highest frequency value or typical value is often used to represent the location of a distribution of values within the larger set of values describing a certain type of variable. Now, the arithmetic mean, or average, is the most commonly used statistic of central tendency, because it provides the most accurate estimate of an expected value given certain assumptions about a distribution (especially symmetry) and requires the least quantity of data for both the accuracy and precision of its result. However, for many real-world collections of data, other less precise statistics (especially the median and the mode) provide a more accurate estimate of the expected value for a given distribution of values. In particular, the median is often used to avoid the effects of extreme outliers in a collection of data, as it is not sensitive to the presence of a few extreme outliers and it is easy to


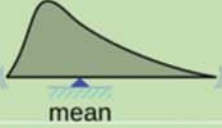
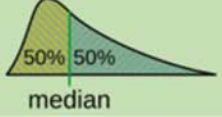
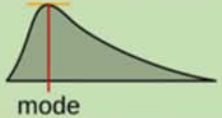
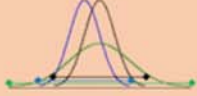
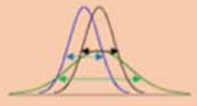
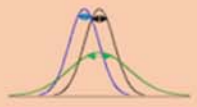
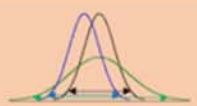

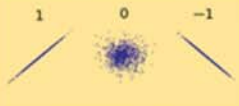

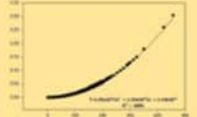
Statistic	Symbol	Formula	Description	Visual description
Size	n	$n = x $	Cardinality of the sample variable (set of data).	
(Arithmetic) Mean	\bar{x} OR μ_x	$\bar{x} = \frac{\sum x}{n}$	Average value of a sample variable. It is the expected value, based on the concept of "central tendency".	
Median	Md	Md = $\frac{x_{n+1}}{2}$ for odd n . Md = $\frac{x_{\frac{n}{2}} + x_{\frac{n}{2}+1}}{2}$ for even n .	The number separating the higher half of a sample variable from the lower half.	
Mode	Mo	No simple formula exists.	The most frequent value for a sample variable.	
Range	range	range = $[\min(x), \max(x)]$	The set of non-repeating values that are observed or possible.	
Variance	σ_x^2	$\sigma_x^2 = \frac{\sum (x - \bar{x})^2}{n - 1}$	Spread of repeatable measured values around the sample mean.	
Standard error (of the mean)	SE _x OR $\sigma_{\bar{x}}$	$\sigma_{\bar{x}} = \frac{\sigma_x}{\sqrt{n}}$	Probabilistic description of how close the sample mean is to the expected value.	
Confidence interval	CL _{x,lc}	CL _{x,lc} = $[x_{(100-lc)/2}, x_{(100+lc)/2}]$	Identifies a range which includes the expected value at some level of confidence, typically 95% or 99%.	
Covariance	Cov(x, y) OR σ_{xy}^2	$\sigma_{xy}^2 = \frac{\sum (x - \bar{x})(y - \bar{y})}{n - 1}$	Describes how two sample variables vary together and can range from $(-\infty, \infty)$.	
Pearson's correlation (linear)	r_{xy}	$r_{xy} = \frac{\sum (x - \bar{x})(y - \bar{y})}{(n - 1)\sigma_x\sigma_y}$	Describes the linear dependence between two sample variables and can range from $[-1, 1]$.	
Spearman's correlation (nonlinear)	r_s OR ρ	Depends on whether repeated values are present.	Describes the monotonic, nonlinear dependence between two sample variables and can range from $[-1, 1]$.	
Coefficient of determination	R^2	$R^2 = 1 - \frac{FUV}{SS_{total}}$ $= 1 - \frac{SS_{residual}}{SS_{total}}$	A measure of how well one or more sample variables fit a given mathematical model, which can range from $[0, 1]$.	

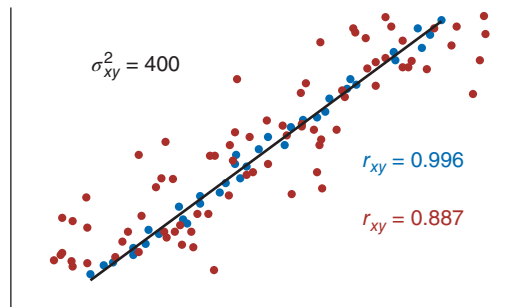
Figure 18.4 Commonly used descriptive statistics for sample variables. Light blue rows are statistics of data quantity. Light green rows are statistics of central tendency. Light orange are statistics of expected intervals. Light yellow rows are statistics of dependence. Source: Reproduced with permission of Cmglee, https://commons.wikimedia.org/wiki/File:Visualisation_mode_median_mean.svg. Licensed under CC By 3.0.

reliably calculate. The mode is very insensitive to various aberrations from commonly expected distributions of values but is often less precise; it requires a much larger quantity of data for a reliable calculation, making it much harder to calculate in a predictable manner.

The next category of descriptive statistics in the light orange rows in Figure 18.4 provide a summary of the observed values in terms of expected intervals and dispersion. The simplest of these statistics is the range that, mathematically, is the set of non-repeating values derived from the collection of all observations (data) or all possible observations, where set is mathematically defined as a collection of “distinct” objects, which are non-repeating values in this situation. However, range has a variety of different but related meanings, including the classical statistical definition, which is the difference between maximum and minimum values from a collection of values that are numerical or ordinal. However, the mathematical definition – a set of non-repeating values – is most useful in the context of a nominal variable. For a numerical or ordinal variable, the range is most often described as the interval encompassing both the minimum and maximum values and is represented as these values separated by a comma between brackets, such as the range [2, 11] for the collection of observations {2, 4, 5, 5, 5, 6, 6, 8, 11} (Galton 1886; Pearson 1895). A parenthesis can be used at either end to indicate up to but not including the boundary value; for example, the notation [0, 10) indicates that the range can include all values from 0 up to but not including 10. This definition of range best captures the concept of an expected interval of values. Sample variance is the next descriptive statistic of expected intervals and represents the spread of measured values around the sample mean. Like the mean, the descriptive accuracy of a variance with respect to an expected interval of values depends on certain assumptions about the underlying distribution of values, especially symmetry. The square root of the variance is the standard deviation (often abbreviated as SD or StdDev), a metric that is an easier quantity to relate to the sample mean. Standard error of the mean (often abbreviated as SE or SEM) is a probabilistic description of the precision of the sample mean with respect to the population mean. The accuracy or confidence of the sample mean can be defined in terms of standard error units under certain assumptions of the underlying distribution of values, especially symmetry. The final descriptive statistic of expected intervals is the confidence interval (CI) that identifies a range, which includes the expected value at some level of confidence. This is a great description of an expected interval, as it makes no assumptions about a distribution and, like the sample mode, is rather insensitive to various aberrations of commonly expected distributions. An alternative formula for calculating the CI based on the sample mean and standard error is $CI_{x,z} = [\bar{x} - zSE_x, \bar{x} + zSE_x]$, where a $z \approx 2$ is equivalent to a 95% CI and a $z \approx 3$ is equivalent to a 99% CI.

The final category of descriptive statistics in the light yellow rows in Figure 18.4 provides a description of dependence between two sample variables. Sample covariance describes how two sample variables vary together and is calculated in a manner analogous to sample variance. In other words, sample covariance describes how the measured values from both sample variables co-spread in a linearly dependent manner around their respective sample means. From one visual perspective, covariance describes an area of co-dispersion centered on the sample means, with the range of $(-\infty, \infty)$. Next, the Pearson’s correlation coefficient, or Pearson’s correlation for short, describes the linear dependence between two sample variables (Pearson 1895). It is related to the covariance by the inverse of the standard deviation of both sample variables and is often viewed as a covariance normalized by the standard deviation of each sample variable. This normalization forces the Pearson’s correlation into a range of $[-1, 1]$, which is often easier to interpret in terms of strength of the dependence between the two sample variables. This interpretability of correlation over covariance is illustrated in Figure 18.5, where the covariances between x and y are the same for two samples, but the Pearson’s correlations for blue and red samples are different. The higher variances present in the red sample represent a lower dependence between x and y even though the nature of the dependence (slope of the black regression line) is the same. This example illustrates why covariance and correlation cannot be quantitatively compared with each other, even though they can be qualitatively compared in terms of their sign: a positive covariance will have a corresponding positive

Figure 18.5 Covariance versus correlation. The red sample has higher sample variances than the blue sample which equates to lower correlation, even though the covariance is the same between the sample variables. Source: Moseley, Hunter (2017): Example of covariance-correlation differences. figshare. doi.org/10.6084/m9.figshare.4968149.v1.



correlation and likewise for negative and zero covariances and correlations. The Spearman's rank correlation coefficient (or Spearman's correlation, for short) describes the monotonic, non-linear dependence between two sample variables (Spearman 1904). A monotonic relationship between two sample variables means that the rank order of the values of the two sample variables is preserved. Visually, a monotonic relationship means that any given horizontal or vertical line will only cross the curve described by the function $y = f(x)$ once, where x and y are the two sample variables. Spearman's correlation describes the strength of this monotonic or rank order dependence between two sample variables in a manner that is analogous to how the Pearson's correlation describes the linear dependence between two sample variables. Specifically, Spearman's correlation is calculated in terms of the preservation of the rank order or inverse rank order between the two sample variables with a range of $[-1, 1]$. The final common descriptive statistic of dependence is the coefficient of determination, which is a measure of how well one or more sample variables fit a given mathematical model. However, from another perspective, the coefficient of determination describes the model-based dependence of a set of sample variables. This statistic is calculated as one minus the ratio of the sum of squares of the residuals over the sum of squares of total differences between observed values and the sample mean. The coefficient of determination can range from $[0, 1]$ and is often described in terms of the fraction of unexplained variance between the model and the data. For a linear model, the coefficient of determination reduces to r^2 , the square of the linear correlation, which is the square of a Pearson's correlation if only two sample variables are involved. More broadly, Pearson's correlation, Spearman's correlation, and the coefficient of determination all measure the strength of dependence between sample variables and some model describing specific mathematical relationships. When there is an expected linear relationship between two sample variables, a Pearson's correlation is typically used to describe the amount of linear dependence. When there is an expected non-linear monotonic relationship between two sample variables, a Spearman's correlation is used to describe the non-linear monotonic dependence. When there is an expectation of a specific mathematical model involving one or more sample variables that is not easily handled by the first two measures of dependence, a coefficient of determination is typically used to describe the dependence of variable(s) with respect to the model.

The Right Graph Is the Most Descriptive Representation of a Dataset

Graphs are simplified drawings that illustrate one or more variables of data within a dataset. In many cases, a graph provides a summative overview of the variables in a visual manner that highlights specific descriptive statistics of the data or properties of a distribution (see the visual descriptions in Figure 18.4). Often, a graph can visualize dependencies between variables, making specific relationships apparent. There are many different types of graphs designed to summarize or highlight sets of variables and even whole datasets in a variety of ways, such as the many charts available in typical spreadsheet software. In most cases, the number of sample variables and/or experiments being organized, analyzed, or visualized together (that is, the data dimensionality) limits which type of graph is usable for a particular visualization task. Given that the vast majority of pictures are two dimensional, most data visualizations in the form of a single graph cannot easily represent more than two dimensions of data directly. But

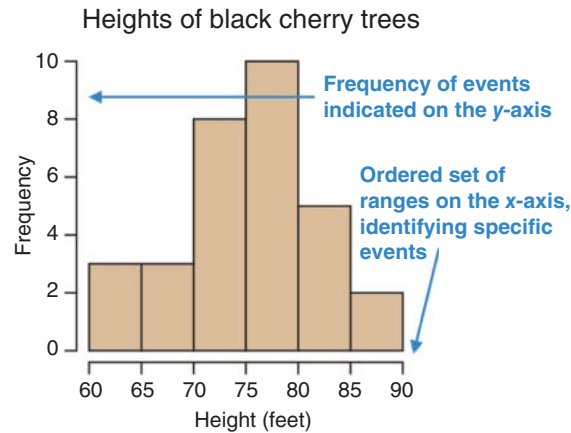


Figure 18.6 Example histogram demonstrating the frequency of black cherry tree heights. Source: commons.wikimedia.org/wiki/File:Black_cherry_tree_histogram.svg CC BY 2.5, commons.wikimedia.org/w/index.php?curid=3483039.

sometimes three dimensions is visualized with depth artificially implemented when a compelling data visualization is needed like representing and comparing volume.

One of the simplest descriptive graphs is the histogram (Figure 18.6), which visualizes the distribution of values for a given sample variable. From a statistical perspective, the histogram visualizes the frequency of an ordered set of statistical events occurring in a sample variable. Most often, each statistical event (i.e. a set of possible outcomes) is represented as a numerical range of possible values, and the height of each bar is the frequency of occurrence of the represented event within the sample variable. Thus, a histogram often provides a clear visual representation of the distribution of values *if* the right ordered set of ranges is used. However, many times, we want to use a graph that enables the visual comparison of a single random variable across multiple experiments. The box-and-whisker plot, or boxplot, was developed exactly for this purpose back in the early 1950s and later popularized by the mathematician William Tukey in the late 1960s (Spear 1952; McGill et al. 1978). Figure 18.7a illustrates the parts of a standard boxplot, which includes the interquartile range (IQR). The bottom of the box defines the first quartile (Q1 or 25th percentile, representing the lowest 25% of the data), the middle indicates the median (Q2 or 50th percentile), and the top of the box defines the third quartile (Q3 or 75th percentile, representing the highest 25% of the data). The whiskers extend from the top and bottom of the box up to 1.5 times the IQR. Any value outside these ranges is often indicated by an outlier point. The boxplot is often used for displaying and comparing distributions of data without making assumptions about the distribution, providing a non-parametric view of the underlying data with respect to rank order and frequency that facilitates visual comparison of datasets. However, the real cleverness of the boxplot was that it enabled effective hand-drawing of descriptive graphs before computers were widely available for this task. But the simplicity of the box and whisker representation is also a drawback. Figure 18.7b shows boxplots that appear to be almost identical (Choonpradub and McNeil 2005). However, the overlay of the data onto the boxplots in Figure 18.7c clearly illustrates how different each sample is. Today, there are several variants of boxplots that are much more descriptive than the original, easy-to-draw boxplot. The violin plot (Hintze and Nelson 1998), the SinaPlot (Sidiropoulos et al. 2018), and their combination (illustrated in Figure 18.7d–f) are visually very descriptive of the different distributions of values and provide a better comparison of the four samples.

The next major type of descriptive graph is a scatter plot representing multidimensional data points that visualize the co-dispersion and dependency between two or more sample variables that are typically quantified using correlation- and covariance-descriptive statistics. Each data point illustrated on the graph represents an ordered set of linked values corresponding to different sample variables; for example, (62.5 in., 101.3 kg) represents the measured height and weight of a male human subject. Often, regression lines or curves are added to the graph to illustrate the dependence of the sample variables with respect to a particular mathematical model or function. Figure 18.8 illustrates the usefulness of scatterplots in four famous

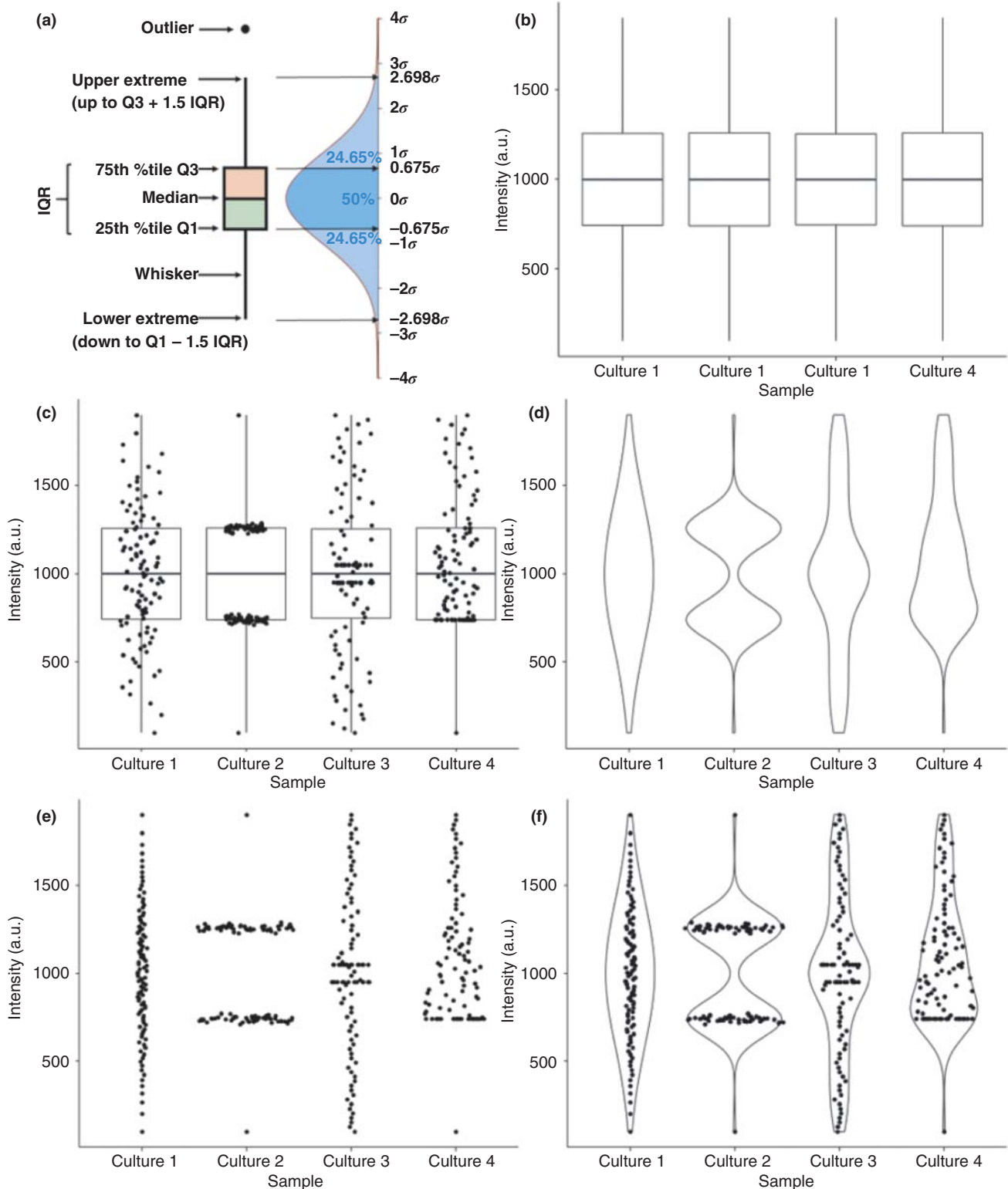


Figure 18.7 Example boxplot and related variant graphs. (a) Schematic diagram of a boxplot. (b) Classical boxplot for four samples of measurements. (c) Classical boxplot with data points. (d) Violin plot. (e) SinaPlot. (f) Violin plot with SinaPlot overlay. Graphs in (b)–(f) were generated using ggplot2 in R. Source: Moseley, Hunter (2017): diagram of a box plot. figshare. doi.org/10.6084/m9.figshare.4993937.v1. Moseley, Hunter; Flight, Robert M (2017): Standard Box Plot. figshare. doi.org/10.6084/m9.figshare.4968152.v1. Moseley, Hunter; Flight, Robert M (2017): Boxplot with data points. figshare. doi.org/10.6084/m9.figshare.4968155.v1. Moseley, Hunter; Flight, Robert M (2017): Example Violin Plot. figshare. doi.org/10.6084/m9.figshare.4968158.v1. Moseley, Hunter; Flight, Robert M (2017): Example SinaPlot. figshare. doi.org/10.6084/m9.figshare.4968161.v1. Moseley, Hunter; Flight, Robert M (2017): Example Violin plot plus SinaPlot. figshare. doi.org/10.6084/m9.figshare.4968164.v1.

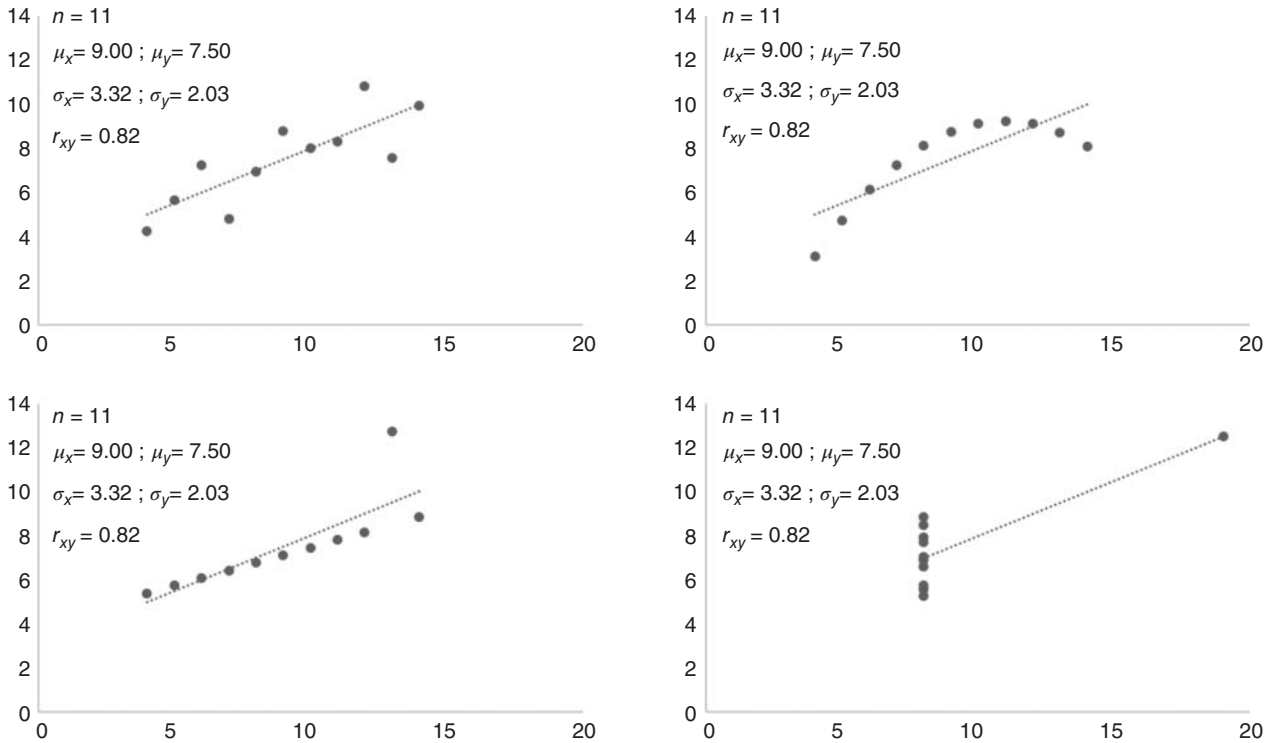
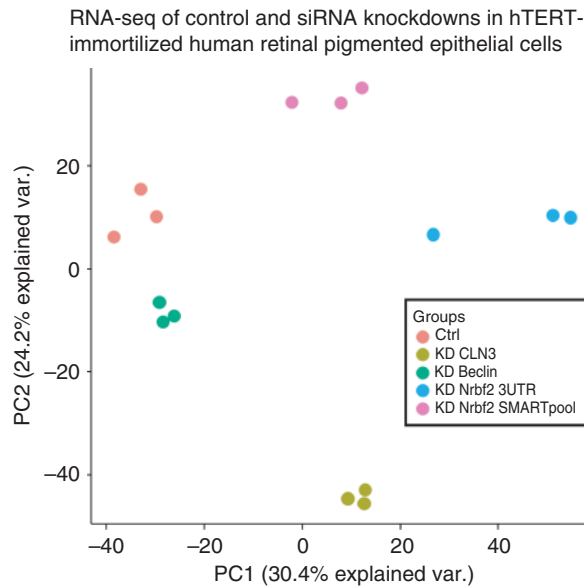


Figure 18.8 Anscombe's quartet. Scatterplots with regression lines for four famous sets of data points that are visually very distinct, but provide the exact same descriptive statistical values.

graphs known as Anscombe's quartet (Anscombe 1973). In the early 1970s, Francis Anscombe created four datasets, each containing two variables that provided identical values for all standard descriptive statistics, including linear regression lines. However, simple scatterplots of the datasets reveal just how different these four datasets really are. What is most disconcerting is the very high linear correlation r_{xy} for two sets of points that clearly do not have the linear relationship indicated by the regression line. The four graphs are a warning not to over-interpret descriptive statistics, especially in the comparison of small datasets and without knowing the nature of the dependency between variables.

Scatterplots have also proven useful in visualizing certain derived summative properties or statistics of high-dimensional datasets that contain hundreds or even thousands of related sample variables. The general approach is to derive a small set of descriptive latent variables (variables not directly observed) from a large set of related sample variables and then visualize this set of latent variables with scatterplots. A very common method used for this approach is principal component analysis (PCA), a method that derives principal components of correlation (typically linear correction) from a set of continuous variables. PCA derives each component of correlation in an order that provides the largest amount of variance in the dataset first, with the first principal component representing the largest amount of variance, the second principal component representing the second largest amount of variance, and so on. The analysis is typically repeated until either the desired number of components for visualization is obtained or until a pre-defined fraction of dataset variance is represented by the resulting list of principal components. Thus, PCA creates the smallest set of latent continuous variables with no correlation between each other that represents the largest cumulative fraction of the variance present in the original high-dimensional dataset. Figure 18.9 shows a PCA scatterplot of two principal components of linear correlation derived from RNA-seq datasets for five groups of human retinal pigmented epithelial cells immortalized with human telomerase reverse transcriptase (RPE-1) and with three replicates in each group. One group is the control and the other four groups represent small interfering ribonucleic acid (siRNA) knockdowns for three

Figure 18.9 Scatterplot of the first two principal components (PCs) from principal component analysis. Source: Moseley, Hunter; Flight, Robert M; Wang, Qingjung (2017): PCA plot of RNAseq dataset of CLN3 knockdown. figshare. doi.org/10.6084/m9.figshare.4994204.v1.



different genes, with one gene knocked down in two different ways. The two principal components contain over 50% of the variance in the combined dataset that includes thousands of random sample variables for individual RNA abundances, which are typically used to infer levels of gene expression. These first two principal components highlight the separation between the five groups of RPE-1 cells and indicate that the difference between the five groups is represented in the largest sources of variance in the combined dataset.

As illustrated by the histograms, the various types of boxplots, and the scatterplots shown in the figures discussed above, graphs can provide very descriptive representations of data. However, care must be taken to make them maximally descriptive. The following points provide useful guidelines for making graphs very descriptive.

- Always include a descriptive title in a graph, such as “Isocitrate dehydrogenase 1 activity.” Also, do not just repeat the axis labels in the title.
- Always label the axes with a descriptive name and the units of measurement; for example, “Culture growth time (h),” “ $\mu\text{g}/\text{ml}$ protein,” or “Intensity (a.u.).”
- Visually represent uncertainty in the data whenever possible and reasonable.
- Using error bars and visualizing the underlying distribution are two major ways to visually represent uncertainty.
- Use error bars that are useful for interpretation. Typically, larger error bars help prevent over-interpretation of data.
- Always identify the units and size of the error bars (e.g. “Error bars represent two SE units”).
- Include legends when multiple datasets, groups, or types of data are present and need to be identified.
- Pick the right graph that does not hide key descriptive characteristics of the data. This may require trying a variety of graphs with different settings to find the graph that is just right.
- Choose one message for each graph and focus on communicating just that message. For instance, a graph may communicate one set of results and associated conclusion.

These guidelines both help an audience understand what a graph is representing and facilitate the interpretation of the underlying data represented in the graph. Figure 18.10 illustrates why these guidelines matter. For instance, Figure 18.10a lacks quite a few descriptive items, including a title, axis title, legend, and error bars, which limits the interpretability of the graph and frustrates the reader. Smartly, Figure 18.10b has all of these visual characteristics, allowing the viewer to quickly decipher what the graph represents and what is informationally important in the graph – which, in this case, is a comparison of wild-type vs. knockout mice with respect to locomotor activity.

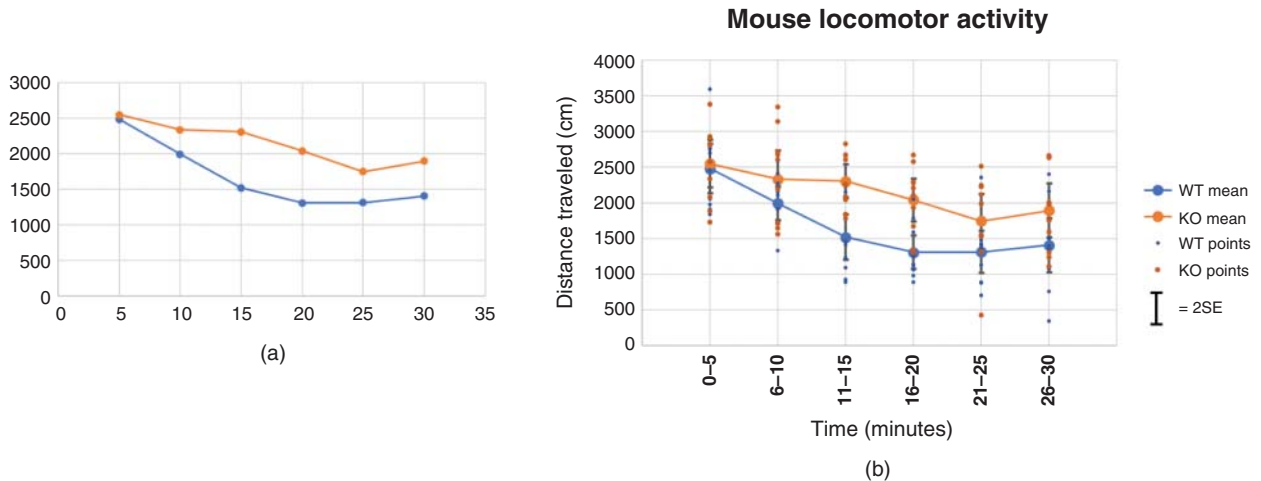


Figure 18.10 Example of how to make a graph descriptive. Source: Moseley, Hunter (2017): Bad and Good Graphing Examples. figshare. doi.org/10.6084/m9.figshare.4994207.v1

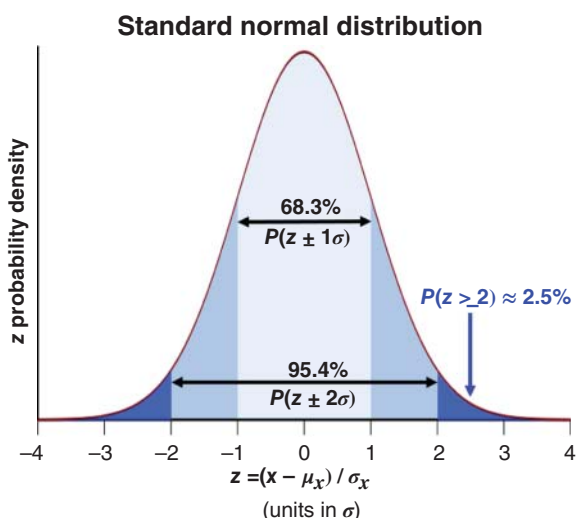
Frequency and Probability Distributions

One of the main purposes of visualizing data is to verify that certain interpretations of descriptive statistics are valid. When the number of observations are relatively few (i.e. less than 100), direct visualization of data is relatively straightforward in most types of plots, including the data visualized using the boxplot and boxplot variants in Figure 18.7c,e,f and the scatterplot in Figure 18.10b. As the number of observations increases above 1000, direct visualization of the data becomes harder. However, a visual description of the distribution of observations is still needed to verify key assumptions often required for specific interpretations of more quantitatively descriptive statistics. Remember that a frequency distribution is the frequency at which specific values within a given set of values occur and a histogram (Figure 18.6) can be a very descriptive visualization of a frequency distribution that summarizes a single ordinal or numerical sample variable of data, especially when the amount of data being summarized is large. The related probability distribution is the set of probability densities at which specific values occur. In the context of a continuous random variable, a probability density is how likely it is that a specific value (outcome) will occur relative to an infinite number of other possible values (outcomes). Also, a probability is the likelihood of an event occurring, where the event is defined as a continuous range of values in this context. Moreover, one can view probability density as the relative frequency that sums up to a total probability of 1 for the whole distribution. Figure 18.11 illustrates the best known and most commonly observed probability distribution: the standard normal distribution, also known as the Gaussian distribution. It is named the Gaussian distribution after Carl Friedrich Gauss, who provided the first specific description of the normal distribution in 1809 (Gauss 1809). In Figure 18.11, the x -axis describes values in terms of a z -score, where $z = (x - \mu_x) / \sigma_x$. Here, the z -score represents the value in terms of a deviation from the mean μ_x that is normalized by the standard deviation σ_x . Thus, the x -axis is in units of standard deviation. The y -axis describes the probability density at specific values of z , which is often described in terms of a probability density function, $y = \text{pdf}(z)$; or $\text{pdf}(x)$ if the variable x is being directly used. Now, the actual probability P of certain statistical events like $\{z \geq a\}$ can be defined as:

$$P(z \geq a) = \int_a^{\infty} \text{pdf}(z) dz$$

This is simply the area underneath the $\text{pdf}(z)$ curve starting at $z = a$. Likewise, the probability of a set of absolute z -values less than or equal to a certain number of standard deviation units

Figure 18.11 The standard normal distribution. Source: Moseley, Hunter (2017): Description of a normal distribution. figshare. doi.org/10.6084/m9.figshare.4994210.v1



can be expressed as:

$$P(-a \leq z \leq a) = \int_{-a}^a \text{pdf}(z) dz$$

For a normal distribution, 68.3% of the probability density lies within $\pm 1\sigma$ and 95.4% lies within $\pm 2\sigma$. However, these expectations of probability are different for other common distributions. As illustrated in Figure 18.12, there are many different well-characterized distributions seen in biological and biophysical data, like the log-normal distribution, the Poisson distribution, and variants of the binomial distribution, especially the negative binomial distribution. However, many collections of related observations represent a summation of several similar but independent distributions that tend to approximate a normal distribution even though the distributions themselves are not normal. While each independent distribution technically represents a different phenomenon, pragmatically, it is not possible to collect them as separate random variables a priori. This tendency for the sum of independent random variables to approximate a normal distribution is known as the central limit theorem, which is a foundational principle of statistical and probability theory. The central limit theorem is also the salvation of many biological and biophysical datasets, since collections of related observations are often a summation of distributions that can be approximated or treated as a normal distribution.

However, real distributions are never as pretty as the ideal statistical model of the distribution. Figure 18.13 shows graphs describing the distribution of certain bond lengths and coordination angles in metalloproteins (Yao et al. 2017). Figure 18.13a illustrates several overlapped histograms of real distributions of bond lengths between specific metal ions and oxygen ligand atoms in metalloproteins. Figure 18.13b displays several overlapped histograms of the smallest ligand–metal–ligand angle for coordinated zinc metal ions. These distributions of bond lengths and coordination angles were derived from three-dimensional, atomic-level representations of metalloprotein structures within entries stored in the Worldwide Protein Data Bank (wwPDB; see Chapter 12) (Berman et al. 2007). Several of these distributions show aberrations from the ideal normal distribution. The most striking aberration is having more than one mode as illustrated by the bimodal green distribution in Figure 18.13b, involving the coordination of a zinc ion by five ligand atoms. Modality or the number of modes present is a very important characteristic to evaluate in real distributions, as most descriptive statistics of central tendency, such as the mean and median, and descriptive statistics of dispersion, like variance, are only quantitatively interpretable from a probabilistic perspective if the distribution is unimodal. However, well-resolved (non-overlapping) modes of a multimodal distribution can be

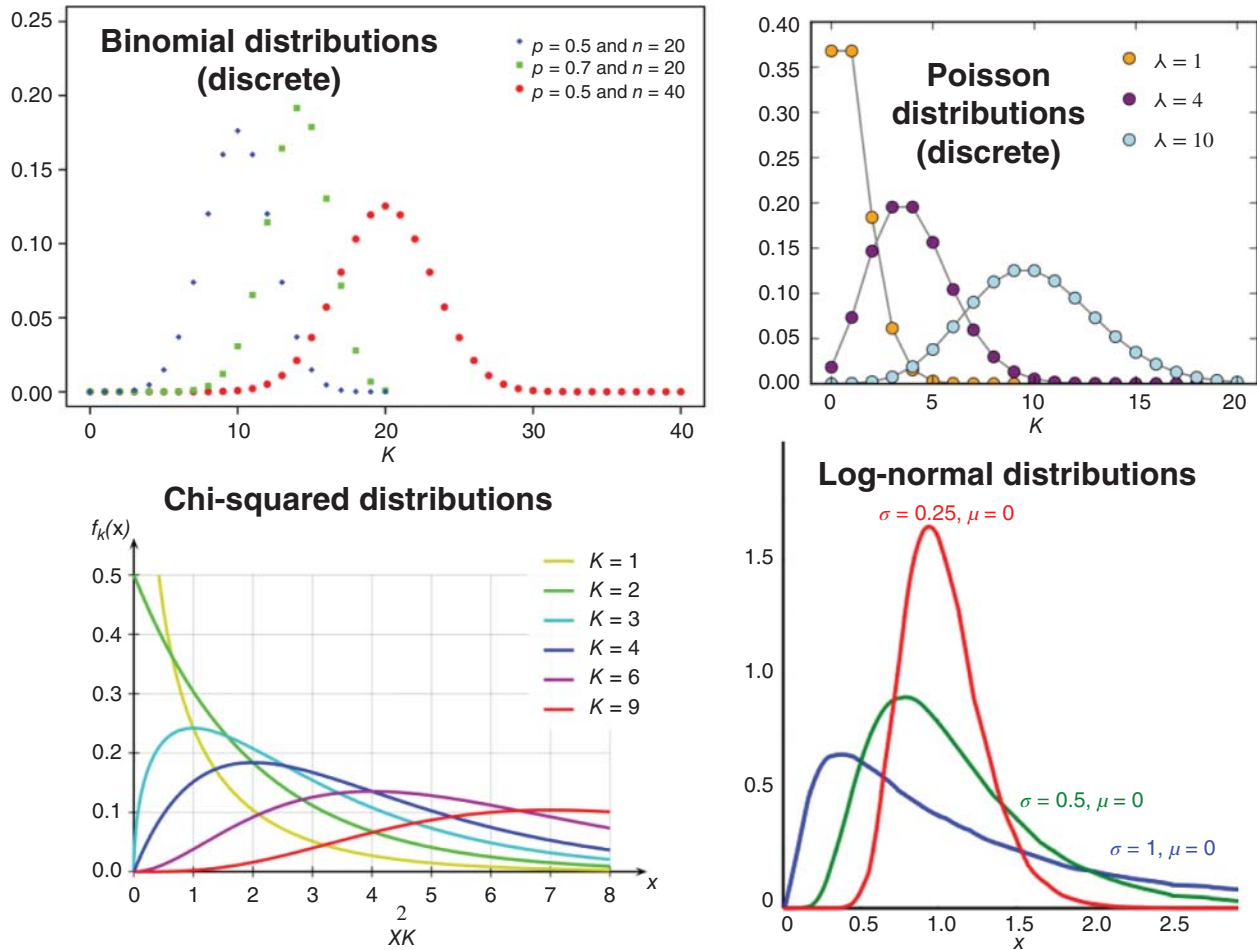


Figure 18.12 Other well-described discrete and continuous distributions commonly observed or used with biological and biophysical datasets. Source: By Skbkekakas – Own work, CC BY 3.0, commons.wikimedia.org/w/index.php?curid=9447142. By Tayste – Own work, Public Domain, commons.wikimedia.org/w/index.php?curid=3646951. By Krishnavedala – Own work, CC0, commons.wikimedia.org/w/index.php?curid=39170496. By Geek3 – Own work, CC BY 3.0, commons.wikimedia.org/w/index.php?curid=9884213.

separated, described, and treated as separate unimodal distributions. Also, the blue distribution in Figure 18.13a is not symmetric owing to an inflation (i.e. higher frequency) of the right tail of the distribution. This deviation from symmetry around the mode of the distribution is called “skewness” and an inflation of the right tail is defined as positive skew. Likewise, the red distribution in Figure 18.13b is also not symmetric, with its left tail inflated in a negative skew. Both multimodality and high skewness can inflate variance and cause serious deviations in the mean and median, limiting the quantitative interpretability of these descriptive statistics.

As demonstrated in several previous figures, histograms and related distribution-descriptive graphs like the violin plot are very useful for visually inspecting the distribution and verifying key assumptions of distributions underlying specific interpretations of the data. However, a minimum amount of data is required to generate these distribution-descriptive graphs. For a histogram, a general rule of thumb is that at least 30 data points are required to represent a unimodal distribution, as illustrated by the 31 points of data visualized in Figure 18.6. But much more data are required to visually characterize other aspects of a distribution such as modality and skewness, especially if modes are not well separated (Figure 18.13). There are also multidimensional distribution-descriptive graphs, including contour plots, that can aid in evaluation of multidimensional distributions. However, these types of graphs require much more data before they become distribution descriptive.

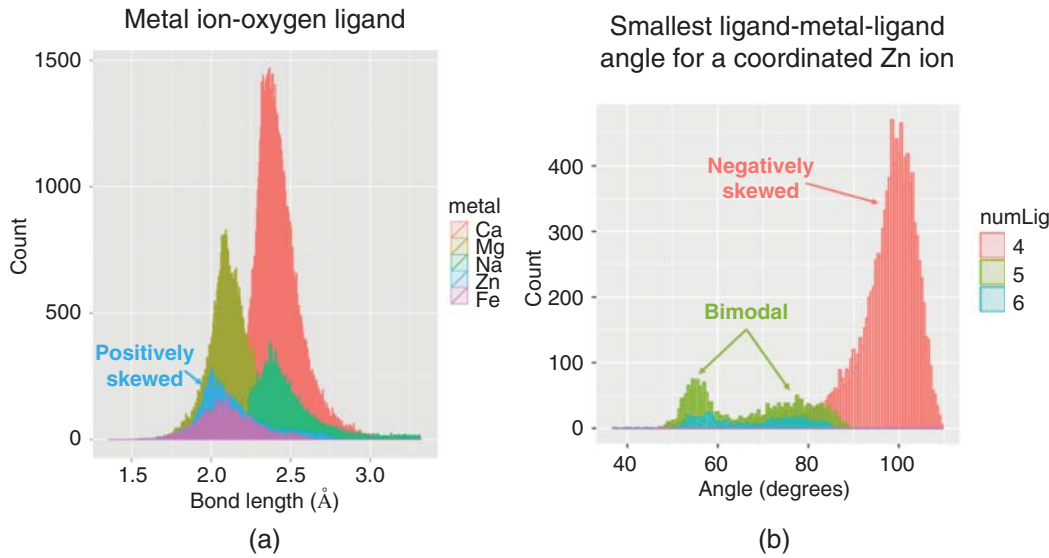


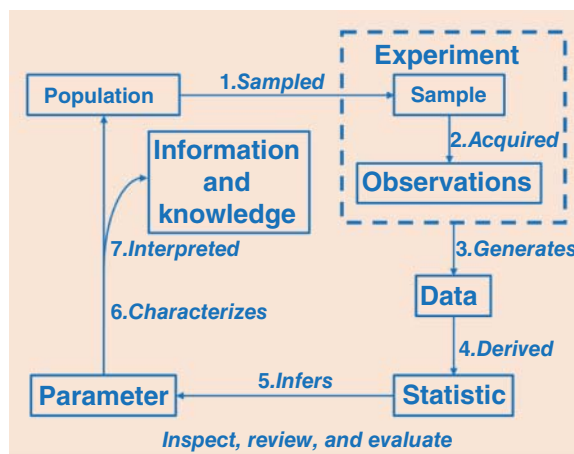
Figure 18.13 Bond length and coordination angle histograms for coordinated metal ions in metalloproteins. (a) Bond length histograms for common metal ions bonded to oxygen ligands in metalloproteins. (b) Smallest ligand–metal–ligand angle histograms for coordinated zinc ions in metalloproteins. Source: Yao, S., Flight, R.M., Rouchka, E.C., and Moseley, H.N.B. (2017). Aberrant coordination geometries discovered in the most abundant metalloproteins. *Proteins: Structure, Function, and Bioinformatics* 85, 885–907. Reproduced with permission of Wiley.

Statistical Inference and Statistical Hypothesis Testing

Statistical Inference

Statistical inference is the process of forming judgments (or “propositions”) about the properties of a population, typically on the basis of (random) sampling, for the overall purpose of gaining new information and knowledge and/or to make informed decisions. Figure 18.14 illustrates this process, starting with a population that is being analyzed. This population is randomly sampled for an experiment where observations are acquired. This generates data that can be used to derive descriptive statistics to infer specific parameters that characterize the underlying population. These descriptive statistics are then interpreted to form new information and knowledge. What is paramount in this process is that the sample, experiment, observations, and data are adequately inspected, reviewed, and evaluated so that derived statistics can be used to infer parameters that accurately characterize the right population(s), allowing

Figure 18.14 Overview of the process of statistical inference. FUV stands for the fraction of unexplained variance. Source: Moseley, Hunter (2017): Overview of a statistical inference process. figshare. doi.org/10.6084/m9.figshare.4994213.v1.



reasonable interpretation that provides new information and knowledge. For example, consider a laboratory that has created a mouse with a gene knockout that produces a very hairy phenotype that they adoringly call the furball. This hairy phenotype is quite unique, so the laboratory breeds 100 of these furballs to produce a random sampling in order to characterize their hairy phenotype. Patches of hair on both the dorsal and ventral side of this sample of mice are measured. The resulting dataset of hair measurements are described using both graphs, which illustrate that the sample distribution approximates a normal distribution and mean and standard deviation-descriptive statistics that appears to completely reproduce the sample distribution. From this sampling, it is inferred that the population of furballs has a hairy phenotype that is normally distributed and well described by their sample mean and standard deviation. The laboratory further validates this result by repeating the random sampling 6 months later, producing very similar inferences about the furball population. For datasets in public scientific repositories, only the data and associated metadata can be adequately inspected, reviewed, and evaluated before downstream analyses. This inspection of publicly archived data is absolutely required, since not every dataset deposited into a repository has had the same level of prior inspection and quality control; in addition, many repositories only require minimal standards for deposition (Brazma et al. 2001). This has led many in the field of bioinformatics to view public scientific repositories as useful and often essential but somewhat “dirty” (Kim et al. 2003). Therefore, many bioinformaticians consider the inspection and removal of unusable data – that is, the “cleaning” of datasets – as the largest single part (and often most critical step) of their work, typically taking ~80% of their effort (Zhang et al. 2003). In reality, inspection, review, and evaluation are simply an underlying part of the overall process of statistical inference, which may require revising or even repeating any given step of the process.

Statistical Hypothesis Testing

The main direct purpose of statistical inference is to form propositions or judgments and statistical hypothesis testing is one of the most common statistical methods used to form these judgments of the data. Within the context of most bioinformatic analyses, a practical definition of hypothesis testing is the comparison of a dataset (sample) with another dataset (sample) or to a model to form a judgment based on the data. However, the more general, statistical definition of hypothesis testing is the creation and testing of a testable hypothesis on the basis of observing a phenomenon that is modeled via a set of random variables. The creation of a testable hypothesis may technically be considered a separate step distinct from statistical hypothesis testing but, often times, the creation of the exact testable hypothesis goes hand in hand with actually testing this hypothesis, as a given hypothesis must be amenable to an available method of testing. Standard implementation of statistical hypothesis testing involves three major steps. The first step involves creating a hypothesis of the form that a statistical relationship between two samples exists. This hypothesis is called the alternative hypothesis (H_a) and it is often directly based on an experimental hypothesis derived from a biological and/or analytical perspective. However, directly testing an alternative hypothesis H_a is often very hard to do. The second step involves creating a logically opposite hypothesis, known as the null hypothesis (H_0), one that is much easier to directly test; in this case, that there is no statistical relationship between two samples. In the third step, one directly tests the null hypothesis H_0 (i.e. the non-existence of the statistical relationship) by comparing values of a statistic derived from each sample. This approach is based on the fact that it is vastly easier to directly falsify a hypothesis or statement than it is to directly prove that a hypothesis or statement is true. Therefore, falsifying a null hypothesis H_0 that is the logical opposite of the desired alternative hypothesis H_a allows the desired alternative hypothesis H_a to be indirectly proven true – but, to understand the null hypothesis H_0 that is directly being tested, it is important to have a clear definition of a statistical relationship in the context of descriptive statistics in order to prevent confusion. When a derived statistic from a given random variable for two samples is not statistically “the same,” this situation is interpreted in terms of the existence of a statistical

relationship between the two samples with respect to this random variable. Thus, “sameness” of the statistic derived from the two samples indicates that a statistical relationship does not exist. For example, consider the null hypothesis H_0 , where the means of the heights for two normally distributed samples of thoroughbred horses are the same, indicating that no relationship exists between the two samples of thoroughbred horses with respect to their height. The lack of statistical sameness (i.e. a statistical difference) would indicate that the null hypothesis is wrong and a statistical relationship between the heights of the two thoroughbred horse samples does exist. Thus, statistically significant differences are used to falsify the null hypothesis H_0 that a relationship does not exist (i.e. the statistic is considered the same for both samples). The main idea to remember is that statistically significant differences are used to falsify (reject) the null hypothesis H_0 that a statistic from two samples are statistically the same in order to confirm an alternative hypothesis H_a that the statistics from the two samples are statistically different and thus the relationship between the two samples exists. These differences in the way that statisticians and biologists perceive, describe, and define the world is a source of a lot of scientific confusion when these two groups of scientists try to communicate with each other, often leading to ineffective or failed collaboration. Therefore, it is very important that the establishment of collaboration across disciplines is done with a lot of patience and a focus on detecting points of miscommunication that often derive from differences in terminology.

Consider the first two steps described above in a statistical hypothesis testing procedure in the context of a biological example. Two sets of cell cultures are grown on plates; one is treated with a drug while the other is not. Samples from both the media and cells of each plate are taken and the relative amount of lactate is measured by a one-dimensional ^1H nuclear magnetic resonance (NMR) experiment after a 24 hour exposure to the drug. The experimentalist is first interested in testing the following experimental hypothesis: “The normalized lactate NMR intensity observations from the media between case and control populations are different.” To test the experimental hypothesis, the following statistical alternative hypothesis H_a is created: the means \bar{x}_a and \bar{x}_b of the two collections of observations from samples S_a and S_b are different. The alternative hypothesis H_a proposes that the relationship (difference in means) between S_a and S_b exists. Next, the logical opposite null hypothesis H_0 is created: the means \bar{x}_a and \bar{x}_b of the two collections of observations from samples S_a and S_b are the same. The null hypothesis H_0 proposes that no relationship between S_a and S_b exists and the rejection of the null hypothesis H_0 during statistical testing will validate the alternative hypothesis H_a , supporting the interpretation that the two sets of cell cultures represent two distinct populations. The statistician in the group asks for a description of each “sample.” The experimentalist starts to describe the “samples” taken from each cell culture. Neither one realizes that the word “sample” means something else to other and the sparks start to fly as they misunderstand what the other is saying.

Type I and II Errors that Arise from Statistical Hypothesis Testing

When testing a null hypothesis H_0 , there is a judgment of whether the test was positive or negative. A rejection of the null hypothesis would support the alternative hypothesis and is considered a positive outcome of the test. A failure to reject the null hypothesis would not support the alternative hypothesis and is considered a negative outcome of the test. However, there are four logical outcomes from the test based on whether the null hypothesis is actually true or false. These outcomes are illustrated by the truth table in Figure 18.15 (see Box 5.4 for additional information). Starting at the bottom left of the truth table, rejecting a null hypothesis that is false is called a true positive, where rejection of the false null hypothesis correctly supports the alternative hypothesis. Moving to the top right of the truth table, not rejecting a null hypothesis that is true is called a true negative, where failure to reject the true null hypothesis correctly does not support the alternative hypothesis. In the top left of the truth table, incorrectly rejecting a null hypothesis that is actually true is called a false positive, where rejection of the true null hypothesis incorrectly supports the alternative hypothesis. In statistics, a false

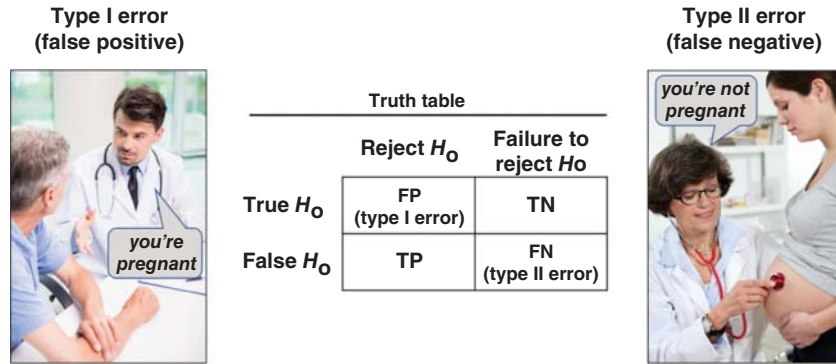


Figure 18.15 Truth table with descriptions of type I and II errors.

positive is called a type I error, where the result of the test leads one to conclude that a statistical relationship exists when in fact it does not. The left image in Figure 18.15 illustrates a type I error when the pregnancy test indicates that the male patient is pregnant. Finally, in the bottom right of the truth table, a failure to reject a null hypothesis that is actually false is called a false negative, where failure to reject the false null hypothesis incorrectly does not support the alternative hypothesis. In statistics, a false negative is called a type II error, where the result of the test leads one to conclude that a statistical relationship does not exist when in fact it does exist. The rightmost image in Figure 18.15 illustrates a type II error when the pregnancy test indicates that the clearly pregnant female patient is not pregnant.

Statistical Significance

The decision to reject the null hypothesis H_0 is not easy to make, especially when one does not know whether the null hypothesis is true or false. The concept of statistical significance helps in this decision-making by framing the judgment in terms of how improbable it is to reject a true null hypothesis, making a type I error. The less likely it is to make a type I error, the more statistically significant the rejection of the null hypothesis is. Figure 18.16 illustrates statistical significance in terms of a probability or p value of obtaining at least as extreme a result as the current null hypothesis H_0 when H_0 is true. For a given H_0 , the p value in green is the

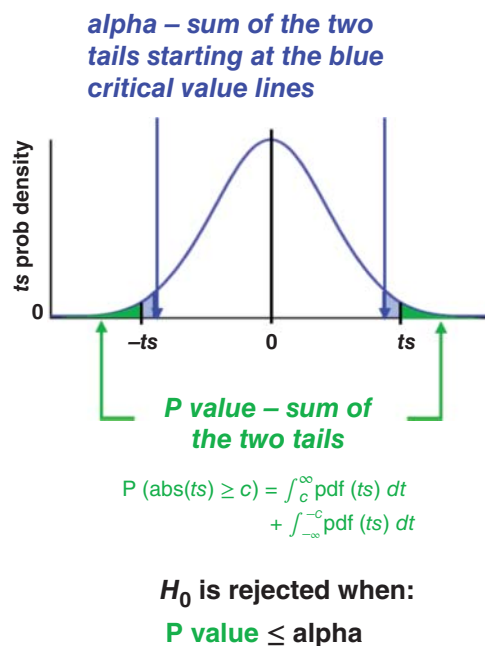


Figure 18.16 Diagram illustrating the relationships between a probability density distribution, p value, and alpha. Source: Moseley, Hunter (2017): Relationships between pdf, p value, and alpha. figshare. doi.org/10.6084/m9.figshare.4994216.v1.

sum of one or two green tails of area under a given probability density function based on the test statistic t . The specific test statistic and whether one tail or two tails are being summed depends on the specific test being performed. Alpha, also called the level of significance, is the probability of rejecting the null hypothesis H_0 when H_0 is true. The alpha (in blue) is calculated as the sum of one or two tails starting at the blue critical value lines. The critical value(s) is simply the test statistic corresponding to a given alpha. The alpha creates the decision point of whether to accept or reject the null hypothesis H_0 based on whether the p value corresponding to the null hypothesis is less than the alpha. Good statistical practice is to pick alpha before performing the statistical test in order to prevent confirmation bias or selecting criteria that help to confirm the desired outcome of an experiment. Typical alphas selected in biological and biomedical research are 0.001, 0.01, or even 0.05 when sample sizes are small.

Testing the Null Hypothesis with a Two-Sample t -Test

Once a null hypothesis has been developed, one should try to test it. This requires that an appropriate statistical test or method be found. For null hypotheses involving the comparison of means between two different samples that are approximately normally distributed, the two-sample (Student's) t -test is an ideal method. The t -statistic illustrated in step 1 of Figure 18.17 is a comparison of the two samples' means divided by the square root of the estimated variance of the mean differences (i.e. the square root of the squared sums of the best estimate of the standard error of each mean). In other words, the separation of the means is evaluated with respect to the uncertainty (variance) in the underlying data used to calculate the means. Instead of using the standard error of each mean as this estimate of uncertainty, a new estimate of variance σ_d^2 is derived from the weighted average of the two samples' variances, σ_a^2 and σ_b^2 , with the assumption that the two variances are two estimates of the same population(s) variance. The t -statistic follows a probability density distribution called a Student's t -distribution. There are actually many t -distributions, as illustrated in step 2 of Figure 18.17, that are related by the parameter ν , which is the number of degrees of freedom in the t -test. In this case, the degrees of freedom are the sum of the two sample sizes n_a and n_b minus 2. In general, degrees of freedom refer to the number of variables affecting the range of possible states of a system and the probability of each state. In the context of statistics, degrees of freedom refer to the number of values that are "free" to vary, affecting the range and probability of outcomes for a given statistic being calculated. In the context of calculating a two-sample t -statistic, degrees

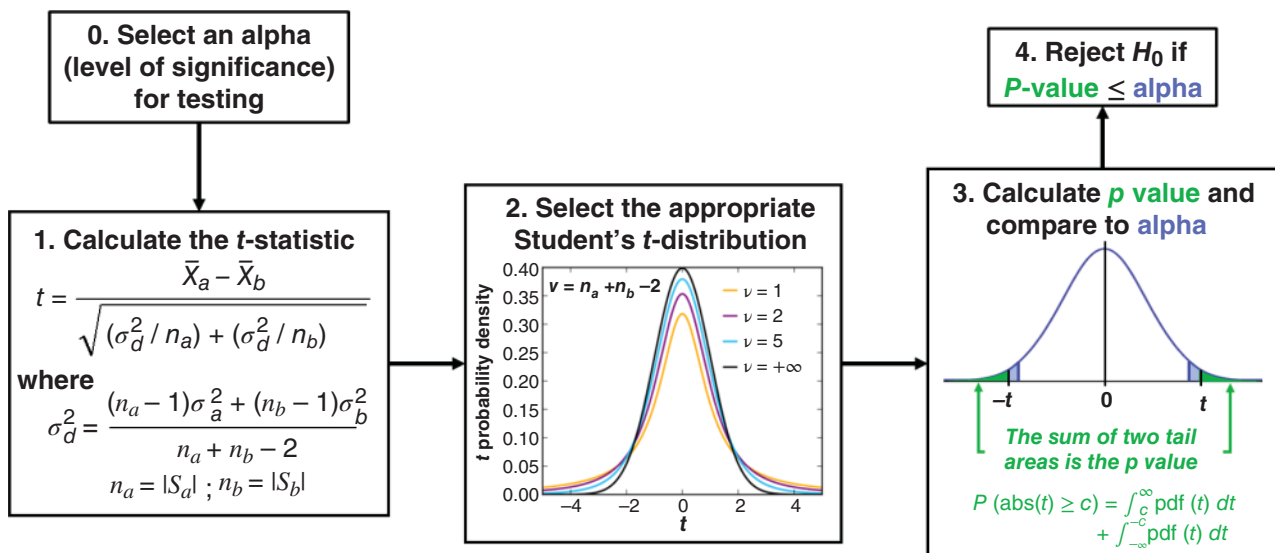


Figure 18.17 Using a Student's t -test to test a null hypothesis. Source: Moseley, Hunter (2017): Overview of using a t -test to test a null hypothesis. figshare. doi.org/10.6084/m9.figshare.4994219.v1.

of freedom refer to the independently sampled collection of observations – values across both samples minus the two mean values which are being directly tested since they are derived from the same two collections of values. Thus, the degrees of freedom dictate the specific Student's t -distribution that is relevant for a specific t -statistic that is calculated. William Sealy Gosset published a description of the Student's t -distributions and their statistical use in 1908 under the pen name Student because of corporate restrictions on publishing by his employer, which is why the Student's t -distribution and Student's t -test are named as they are (Student 1908).

Figure 18.17 illustrates the use of the two-sample t -test in testing a null hypothesis. The process starts with selecting a level of significance, alpha, before performing the test. Then, in step 1, the t -statistic and related statistics are calculated. In step 2, the appropriate Student's t -distribution is selected based on the degrees of freedom v . In step 3, the p value is calculated based on the t -statistic and the probability density function of the appropriate Student's t -distribution. Finally, in step 4, the decision is made of whether to reject the null hypothesis H_0 or not, based on whether the p value is less than or equal to alpha. Normally, all of these steps are done by a statistical t -test function that can be found in many spreadsheet programs and almost all general data analysis packages. All one has to do is provide the two samples of values in the appropriate format, where the two samples are expected to have the same variance and come from a population(s) that is approximately normally distributed.

To better understand what the two-sample t -test is actually doing, an understanding of the relationship between the population distribution and the distribution of possible sample means, created by multiple random samplings of the population, is needed. This is illustrated in Figure 18.18, with the large lighter blue and red population distributions in relation to the darker and much smaller sample mean distributions. Because of the central limit theorem, the smaller sample mean distribution approximates a normal distribution *no matter what the population distribution looks like* when the sample size of the random samplings is sufficiently large. This is because each sample value represents an independent random variable, which are summed in order to calculate a mean statistic that, by the

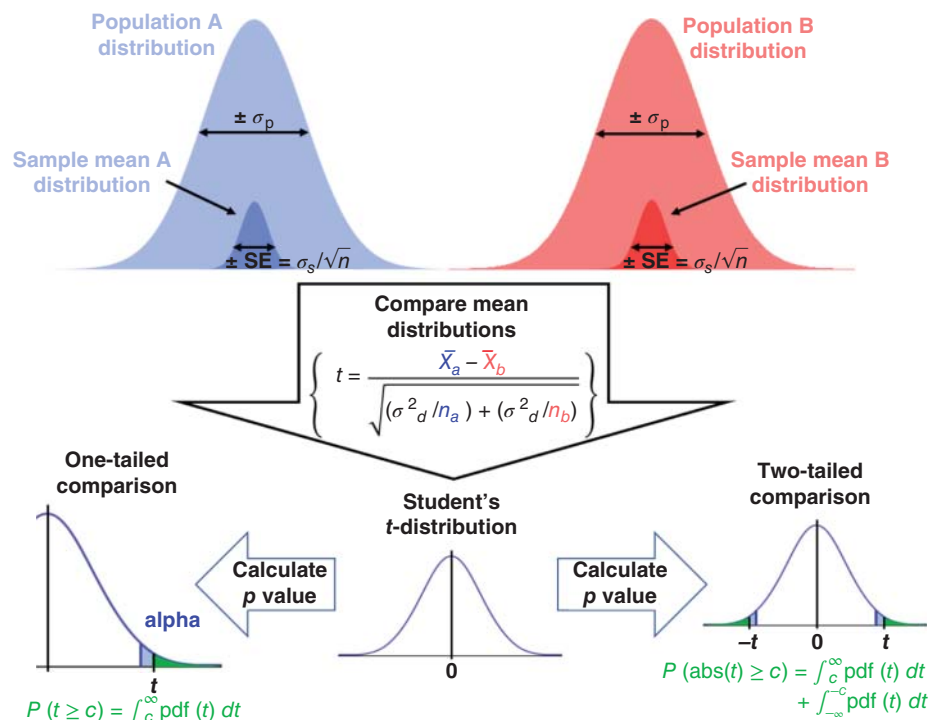


Figure 18.18 Relationship between population and sample mean distributions. Source: Moseley, Hunter (2017): Relationships between population and sample mean distributions. figshare. doi.org/10.6084/m9.figshare.4994222.v1.

central limit theorem, states that the resulting variable (the mean, in this instance) should be approximately normally distributed. That said, how well the mean and standard error of a given sample provide a reasonable estimate of the mean and standard deviation of the sample mean distribution is dependent on how well the original population distribution approximates a normal distribution. So, the distribution of the original population does matter in the context of a two-sample t -test, where the estimates of mean and standard deviation values of the sample mean distribution are coming from only two samples. Now, the comparison of two sample means in the t -statistic can be used to compare pairs of means from the two mean distributions. The resulting probability density distribution of t -statistics is the representative Student's t -distribution. This t -distribution can then be used in either a one-tailed or two-tailed comparison for calculating p values, deciding whether to reject a null hypothesis H_0 . The difference in the one-tailed and two-tailed comparison is whether a single direction of deviation is being tested versus either direction of deviation being tested. For example, whether a drug produces a positive response in the observations of a case-control experiment can be tested with a one-tailed t -test. However, a two-tailed test is used if any significant deviation in either direction between the means of two samples is being tested. Other considerations for the correct selection of statistical hypothesis test are discussed later in this chapter in the Common Statistical Tests Used in a Typical Statistical Inference Process section.

Statistical Power

As previously described, statistical significance focuses on alpha or the probability of type I errors (false positives). What about the probability of type II errors (false negatives)? The statistical term “beta” represents the probability of type II errors and one minus beta represents the concept of statistical power or the probability of correctly rejecting the null hypothesis H_0 . Statistically powerful experiments have a high probability of rejecting actual false null hypotheses – and this is why most reviews of submitted grant proposals for biomedical and clinical research include an evaluation of the statistical power of the proposed experiments, in order to assess the likelihood of success of the proposed research. This evaluation of statistical power requires an estimation of statistical power, and this estimation of statistical power is derived from an analysis of statistical power based on known or estimated statistics. A power analysis relates four interdependent factors within the context of a particular statistical test. These factors are alpha, beta (or $1 - \beta$), sample size, and effect size. Effect size is the quantitative measure of the strength of a phenomenon. Given any three of these factors, the fourth factor can be derived through a power analysis. In many cases, certain factors – especially effect size – are not known and can only be estimated, which means that the fourth derived factor is only an estimate as well. Figure 18.19 illustrates the relationships between these four factors in a power analysis within the context of a Student's t -test in an approximate diagram that has deviations with small sample sizes. As shown in the figure, the effect size is the difference in the means of the sample mean distributions and is typically estimated from the difference of two sample means from an equivalent experiment or “reasonably approximated” from a similar experiment or pilot experiment. The effect size is evaluated in terms of the variance of the mean differences that, in turn, is derived from the variance of the two sample mean distributions. The variances of the sample mean distributions are estimated from the square of the standard error, which is dependent on the sample sizes. Alpha and beta are related around a particular t -statistic critical value that is dependent on the variance of the mean differences; this variance is ultimately dependent on the sample sizes. Therefore, when one of these factors changes, the other factors change as well. Typically, power analyses are used to estimate sample size or statistical power. When estimating a minimal required sample size, a desired statistical power such as 0.9 (90% power) at a given alpha such as 0.01 with a reasonable estimate of effect size are provided. Likewise, when the statistical power of a proposed experiment

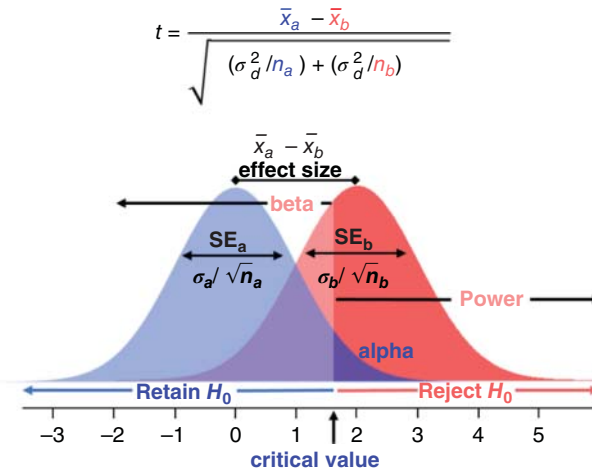


Figure 18.19 An approximate power analysis diagram for a Student's t -test. Source: Moseley, Hunter (2017): Approximate power analysis diagram for a Student's t -test. figshare. doi.org/10.6084/m9.figshare.4994228.v1.

is estimated, an expected sample size at a given alpha like 0.01 with a reasonable estimate of effect size are provided.

Correcting for False Discovery due to Multiple Testing

A common and growing issue in bioinformatics is dealing with experiments containing large numbers of distinct random variables of observations often generated from high-throughput analytical instrumentation like next generation sequencers. While the resulting datasets have various logistical issues related to storage and management of the data, they also pose unique problems for statistical testing and analysis. Probably the most impactful issue comes from the fact that the testing of a large number of random variables will lead to false discovery unless steps are taken to reduce false discovery. For example, consider an RNA-seq experiment involving human cancer cell culture samples that either have or have not been treated with an anti-cancer drug, where the abundance of 7500 unique transcripts across the statistical sample of cell cultures has been measured. Data analysts who are familiar with these datasets wisely explore the data to verify that they behave as expected before moving to the interpretation stage. They do this by graphing the distribution of representative random variables to verify that they appear to approximate a log-normal distribution. They then log-transform the raw gene expression observations and graphs again to see that the resulting distribution is approximately normal. Next, the data analyst selects an alpha of 0.01 and performs a two-sample t -test on each and every random variable of gene expression using the case and control samples. From the testing, 150 of the transcripts pass, but there is a problem. Based on an alpha of 0.01 and the 7500 individual tests performed, 75 (0.01×7500) of the results are expected to be false positives (type I errors). However, only 150 transcripts passed the test. So, roughly 75/150 (or 50%) of the results are false positives, meaning that the rate of false discovery is 50%. Would anyone trust results that are estimated to be 50% incorrect? This example illustrates the fundamental problem with multiple testing of datasets with large numbers of random variables.

The solution to the multiple testing problem is to correct the resulting p values generated from the set of tests being performed. This statistical procedure is called multiple testing correction and its purpose is to limit the false discovery rate, or FDR (see Box 5.4), which is calculated by the following equation based on false positives (FP) and true positives (TP):

$$\text{FDR} = \frac{\text{FP}}{\text{TP} + \text{FP}}$$

During the procedure, a q value or FDR-adjusted p value is calculated based on a p value or group of p values (see Chapter 10). The simplest of the multiple testing correction methods is

the Bonferroni correction, which multiplies a given p value by the number of tests performed to generate an adjusted p value. For a small number of tests, the Bonferroni correction is quick and reasonable, minimizing type I errors. However, the Bonferroni correction method is a very severe correction that often over-compensates in the prevention of type I errors and the lowering of the FDR while severely raising the probability of type II errors (McIntyre et al. 2000). This is because the Bonferroni correction actually controls what is called the family-wise error rate (FWER), the probability of making at least one type I error in the family of tests based on a given alpha. Thus, the Bonferroni adjusted p value is technically not a q value. For many statistical analyses involving high-dimensional datasets, the Bonferroni correction is not a viable multiple correction method owing to the large increase in type II errors caused by the correction of the FWER for hundreds or more tests that can wipe out all statistical significance in the resulting q values. There are other more sophisticated multiple testing correction methods which provide a better balance between limiting type I versus type II errors while achieving a given FDR; however, they are not as easy to use. One of the most popular multiple correction methods is the Benjamini–Hochberg correction that calculates q values by adjusting each p value based on a target FDR and the ordered collection of p values generated from a set of related statistical tests (Hochberg and Benjamini 1990).

The Global Problem with the Use of p Values

Even with multiple testing correction and simply being careful, observant, and conscientious, there is a global problem with the use of p values. First, p values are poorly understood and often misrepresented even in the general scientific community. More often than not, p values are confused with effect size. The measurable strength or size of a phenomenon is not the statistical significance of the detection of the phenomenon from a statistical test. In many cases, once the presence of a particular phenomenon is established, the effect size caused by the detected phenomenon is often more important in evaluating the potential application of the phenomenon. For example, in large genome-wide association studies (GWAS), many statistically significant disease-relevant nucleotide variants have been detected with p values below 10^{-8} ; however, a majority of these nucleotide variants have a disease risk odds ratio below 1.5 (Ku et al. 2010). In this context, the odds ratio is an effect size representing a relatively small increase in disease risk, even though the detected disease association is extremely likely (meaning statistically significant). Likewise, other statistics, such as the E values from BLAST searches that represent the expected number of search hits of the same level of similarity that would occur by random chance for a given sequence database are often confused with p values (see Chapter 3). While p values are consistently calculated, E values are dependent on the database used in their calculation, limiting their interpretation. Second, misinterpretation of p value significance pollutes our scientific literature with significant false discovery. Many published scientific studies have defined “significant” p values in terms of weak alphas (i.e. 0.05). Given the huge number of tests being performed in every scientific laboratory across the world, the selection of significance based on an alpha of 0.05 generates a lot of published false discovery across the combined scientific literature. This published false discovery can be reinforcing when many others try to reproduce the same false discovery they have seen in the published scientific literature. Solutions to this global p value problem are not easy. One journal has even taken the policy of banning the use of p values in its published articles (Woolston 2015). Also, certain scientific communities have worked together to establish guidelines that minimize false discovery. For example, the physics community generally waits to accept major results until five sigmas of significance are reached. This equates to a two-tailed p value of 6×10^{-7} . For human GWAS, p values less than 5×10^{-8} is the standard for accepting results, which is based on a Bonferroni correction of an alpha of 0.05 assuming the presence of 1 million testable independent variants in the human genome (Risch and Merikangas 1996). The field of data science seeks to better understand this phenomenon and model it to create a more robust measurement of significance that does not limit discovery. A good start is to report q

values or other adjusted p values that prevent false discovery in published results, especially for published results involving high-throughput analytical techniques and other experimentation that generates high-dimensional data.

Common Statistical Tests Used in a Typical Statistical Inference Process

There are several widely used statistical tests that are applicable to a wide range of experimental results. Most statistical tests are divided into two categories: parametric tests and non-parametric tests. A parametric test assumes that the sample data come from a population that follows a probability distribution defined by a fixed set of parameters. The most common parametric tests assume the population follows or mimics a normal distribution. A non-parametric test makes no assumptions about the probability distribution of the population or sample. Many of the non-parametric tests like the Wilcoxon–Mann–Whitney test (Wilcoxon 1945; Mann and Whitney 1947) compare the rank order of the samples in a manner that makes no assumptions about the underlying distribution of the population or sample. Table 18.1 provides a list of common parametric tests, their non-parametric equivalents, and the particular statistical situation for their use on a single continuous random variable. Care must be taken that the appropriate statistical test is selected for a given statistical inference. To help in this selection, ask the following four situational questions.

- 1) Does the population(s) or sample(s) approximately follow a normal distribution?
- 2) How many samples (collections of observations/values) are being directly compared?
- 3) Are sample variances or sizes significantly unequal?
- 4) Are observations/values between samples repeated or linked in some way?

The answer to these four questions will help identify the particular statistical situation and the appropriate statistical test for a single continuous random variable. With respect to the first question, if the related population(s) or sample(s) appears normally distributed, then parametric tests are more appropriate and provide superior statistical power and performance. However, if the population(s) or sample(s) has significant deviations from a normal distribution, then a less assuming non-parametric test is most appropriate and provides a better estimate of significance. Graphing each sample using a histogram, when a sample has 30+ values, is a good and reasonably quick way to answer this first question. Moreover, if the distribution appears to be log-normal, then a simple logarithmic transformation of the values may allow the use of a parametric test, and the assumptions of normality can be verified by graphing the transformed sample data. With respect to the second question, whether one sample, two samples, or more than two samples are being tested dictates the specific type of statistical test. For normally distributed data, the t -test and its variants are used when there are one or two samples. When there are more than two samples, the analysis of variance (ANOVA) tests and its variants are used to test whether at least one sample is significantly different than the other samples. Likewise, for significantly non-normally distributed data, the Wilcoxon–Mann–Whitney test and variants are used for testing one or two samples. The Kruskal–Wallis test (Kruskal and

Table 18.1 Common parametric statistical tests and their non-parametric equivalent.

Statistical situation	Parametric	Non-parametric
1 sample	1-sample t -test	1-sample Wilcoxon rank sum
2 samples	2-sample t -test	Wilcoxon–Mann–Whitney test
2 samples, unequal σ^2 , n	Welch unequal σ^2 t -test	Wilcoxon–Mann–Whitney test
Matched pair of samples	Paired t -test	Wilcoxon signed rank test
>2 samples	One-way ANOVA	Kruskal–Wallis test
>2 samples, unequal σ^2 , n	Welch ANOVA	Kruskal–Wallis test
Matched, >2 samples	Repeated measures ANOVA	Friedman test

The two parametric tests in red are appropriate for repeated measures and matched experimental designs and provide the strongest statistical power.

Wallis 1952) is appropriate for testing more than two samples when one of the samples is significantly deviated from normality. With respect to the third question, when two samples are normally distributed but have variances or sizes that differ by more than twofold, the Welch unequal variance t -test (Welch 1947) is very appropriate (Delacre et al. 2017). This modified version of the two-sample Student's t -test compensates for problems caused by disproportionate variances and sample sizes, providing robust statistical performance. The Welch ANOVA is the Welch equivalent for the ANOVA test when more than two normally distributed samples are being compared. The fourth question identifies whether specific observations between samples are linked in a statistically meaningful way. The strongest type of linkage comes from repeated measures experimental designs, where the same biological units or subjects are used to measure observations for each sample, including the control sample. The resulting dataset has linked observations across samples that allows the use of the most powerful statistical tests that directly test the summative statistics of the differences between the linked observations and not the differences of summative statistics between samples. From a biological perspective, the variance between biological units is ignored as only measurements from the same biological unit are compared. For example, comparing the physical performance of mice before and after a treatment allows only the differences between samples of measurements from an individual mouse to be tested, reducing the inclusion of biological variance that is inherent when making comparisons between different mice. When two samples have linked observations, the paired t -test is most appropriate when samples are normally distributed and the Wilcoxon signed rank test is most appropriate when at least one of the samples is significantly deviated from normality. When three or more samples have linked observations, the repeated measures ANOVA is most appropriate when samples are normally distributed and the Friedman test is most appropriate when at least one of the samples is significantly deviated from normality. Besides repeated measurements, less stringent linkage between observations across samples is sometimes derived from matched experimental designs; however, these types of experimental designs have statistical issues that must be addressed and have been criticized as being biased. The related randomized block experimental design is generally accepted as more robust, but requires that biological units be blocked and randomly measured across blocks based on specific potential confounding factors such as age, sex, genetic factors, and even environmental factors like smoking status. These experimental designs generate complex sets of samples that require more complicated statistical tests like the multiple factor (multi-way) ANOVA. Under these circumstances, advice from a statistician is often required both in creating a good experimental design and related experimental procedures and in the selection of the appropriate statistical test(s). This advice should be sought before the experiment is attempted; otherwise, a lot of time, effort, and resources may be wasted generating datasets that are inadequate for the questions being asked. Likewise, when discrete or ordinal random variables need to be statistically tested, consult with a statistician, since there is no general consensus on which statistical tests are appropriate in a given situation and current published recommendations require expert knowledge to interpret (Fagerland et al. 2011). Moreover, multivariate statistical analysis methods are required to simultaneously test multiple random variables. A chi-squared test is one such method that is used, but assumes that the set of random variables are independent and normally distributed. To analyze large numbers of random variables simultaneously, specialized methods like PCA, discriminant analysis, and newer machine learning methods are required. Expert knowledge is needed for their appropriate use and interpretation, given the diverse assumptions about the data each technique makes. Again, seek out advice from statistical and computational experts before blindly using these methods. Go back for additional advice when experimental issues arise. Putting the selection of the appropriate statistical test or method within context, the following steps describe a typical statistical inference process that uses statistical hypothesis testing.

- 1) State what is being tested in terms of a testable (rejectable) hypothesis.
 - State both the null hypothesis H_0 and alternative hypothesis H_a .
- 2) Derive appropriate descriptive statistics and descriptive visual representations of the sample data.
- 3) Evaluate the quality of the data and associated metadata.

- 4) Determine which statistical test is appropriate for the sample data.
 - Seek advice when the appropriate statistical test is not obvious.
- 5) Perform the test and determine its significance (p value).
- 6) If necessary, correct for multiple testing (q value).
- 7) Carefully interpret the results (infer) with respect to the population(s) being studied.

Summary

The modern biological and biomedical research environment has grown data rich and data intensive, requiring that everyday scientists develop data analysis and statistical skill sets in order to effectively analyze, utilize, and interpret the data they generate within the context of global scientific knowledgebases and data repositories. This chapter is meant to be an introductory primer to these skills with a focus on the most important aspects of data interpretation: truly seeking to comprehend and understand the datasets being analyzed using descriptive representations of the data and being methodical and careful in the development, testing, and interpretation of statistical hypotheses of the data. This chapter should be looked upon as a starting point for acquiring statistical knowledge and intuition and not an end-goal. The effective interrogation of data requires both expert knowledge and experience in order to avoid misinterpretation. It takes time to acquire the combined biological, statistical, and computational knowledge and associated experience. Likewise, it is important to recognize when a particular dataset outstrips your current knowledge and expertise and then seek statistical, computational, or analytical advice with patient communication. Also, become an active participant in the collaboration by reading about your colleague's discipline, in order to improve the effective communication required for a successful multidisciplinary collaboration.

Acknowledgments

The author thanks Robert M. Flight for help in the creation of Figures 18.7 and 18.9. The author thanks Qingjun Wang for providing the data illustrated in Figure 18.9.

Internet Resources

Description of histograms, their usefulness, and how to create them	www.cqeademy.com/cqe-body-of-knowledge/continuous-improvement/quality-control-tools/histograms
Description of the pros and cons of different experimental designs	www.simplypsychology.org/experimental-designs.html
Different graphs with almost identical descriptive statistics	www.autodeskresearch.com/publications/samestats
Discussion about boxplots and multimodal distributions	stats.stackexchange.com/questions/137965/box-and-whisker-plot-for-multimodal-distribution/137982#137982
ggplot – libraries in R and Python based on the Grammar of Graphics	pypi.python.org/pypi/ggplot ; cran.r-project.org/web/packages/ggplot2/index.html
plotly – web- and scripting-based platform for data analysis and visualization	plot.ly
R tutorial for basic statistics and graphing	www.statmethods.net
Two-day introduction to R and Bioconductor for transcriptomics analysis	www.bioconductor.org/help/course-materials/2016/BiocIntro-May
Website for the <i>R for Data Science</i> book by Hadley Wickham and Garrett Golemund	r4ds.had.co.nz

Further Reading

- Daniel, W.W. and Wayne, W.D. (1995). *Biostatistics: A Foundation for Analysis in the Health Sciences*. New York, NY: Wiley. A reference manual for common statistical methods used in the health sciences.
- Van Belle, G. (2011). *Statistical Rules of Thumb*, vol. 699. Hoboken, NJ: Wiley. Provides common-sense rules and guidelines to follow when statistically analyzing or interpreting data.
- Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Dordrecht, Netherlands: Springer. A reference manual and handbook for using the grammar of graphics plotting library in R.
- Wickham, H. and Grolemund, G. (2016). *R for Data Science*. Sebastopol, CA: O'Reilly Media. An introduction to structuring, transforming, visualizing, and modeling data using R.

References

- Anscombe, F.J. (1973). Graphs in statistical analysis. *Am. Stat.* 27 (1): 17–21.
- Berman, H., Henrick, K., Nakamura, H., and Markley, J.L. (2007). The worldwide Protein Data Bank (wwPDB): ensuring a single, uniform archive of PDB data. *Nucleic Acids Res.* 35 (Database issue): D301–D303.
- Brazma, A., Hingamp, P., Quackenbush, J. et al. (2001). Minimum information about a microarray experiment (MIAME)—toward standards for microarray data. *Nat. Genet.* 29 (4): 365–371.
- Choonpradub, C. and McNeil, D. (2005). Can the box plot be improved? *Songklanakarin J. Sci. Technol.* 27 (3): 649–657.
- Daniel, W.W. and Wayne, W.D. (1995). *Biostatistics: a Foundation for Analysis in the Health Sciences*. New York, NY: Wiley.
- Delacre, M., Lakens, D., and Leys, C. (2017). Why psychologists should by default use Welch's t-test instead of Student's t-test. *Int. Rev. Soc. Psychol.* 30 (1): 92–101.
- Fagerland, M.W., Sandvik, L., and Mowinckel, P. (2011). Parametric methods outperformed non-parametric methods in comparisons of discrete numerical variables. *BMC Med. Res. Methodol.* 11 (1): 44.
- Galton, F. (1886). Regression towards mediocrity in hereditary stature. *J. R. Anthropol. Inst.* 15: 246–263.
- Gauss, C.F. (1809). *Theoria motus corporum coelestium in sectionibus conicis solem ambientium auctore Carolo Friderico Gauss*. Hamburg, Germany: Sumtibus Frid. Perthes et I.H. Besser.
- Hintze, J.L. and Nelson, R.D. (1998). Violin plots: a box plot-density trace synergism. *Am. Stat.* 52 (2): 181–184.
- Hochberg, Y. and Benjamini, Y. (1990). More powerful procedures for multiple significance testing. *Stat. Med.* 9 (7): 811–818.
- Kim, W., Choi, B.-J., Hong, E.-K. et al. (2003). A taxonomy of dirty data. *Data Min. Knowl. Disc.* 7 (1): 81–99.
- Kruskal, W.H. and Wallis, W.A. (1952). Use of ranks in one-criterion variance analysis. *J. Am. Stat. Assoc.* 47 (260): 583–621.
- Ku, C.S., Loy, E.Y., Pawitan, Y., and Chia, K.S. (2010). The pursuit of genome-wide association studies: where are we now? *J. Hum. Genet.* 55(4), 195–206.
- Mann, H.B. and Whitney, D.R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *Ann. Math. Stat.* 18: 50–60.
- McGill, R., Tukey, J.W., and Larsen, W.A. (1978). Variations of box plots. *Am. Stat.* 32 (1): 12–16.
- McIntyre, L.M., Martin, E.R., Simonsen, K.L., and Kaplan, N.L. (2000). Circumventing multiple testing. *Genet. Epidemiol.* 19 (1): 18–29.
- Pearson, K. (1895). Note on regression and inheritance in the case of two parents. *Proc. R. Soc. Lond.* 58: 240–242.

- Risch, N. and Merikangas, K. (1996). The future of genetic studies of complex human diseases. *Science* 273 (5281): 1516–1517.
- Sidiropoulos, N., Sohi, S.H., Pedersen, T.L. et al. (2018). SinaPlot: an enhanced chart for simple and truthful representation of single observations over multiple classes. *J. Comput. Graph. Stat.* 27 (3): 673–676.
- Spear, M.E. (1952). *Charting Statistics*. New York, NY: McGraw-Hill.
- Spearman, C. (1904). The proof and measurement of association between two things. *Am. J. Psychol.* 15 (1): 72–101.
- Student (1908). The probable error of a mean. *Biometrika* 6 (1): 1–25.
- Welch, B.L. (1947). The generalization of ‘Student’s’ problem when several different population variances are involved. *Biometrika* 34 (1/2): 28–35.
- Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biom. Bull.* 1 (6): 80–83.
- Woolston, C. (2015). Psychology journal bans P values. *Nature* 519: 9.
- Yao, S., Flight, R.M., Rouchka, E.C., and Moseley, H.N. (2017). Aberrant coordination geometries discovered in the most abundant metalloproteins. *Proteins* 85 (5): 885–907.
- Zhang, S., Zhang, C., and Yang, Q. (2003). Data preparation for data mining. *Appl. Artif. Intell.* 17 (5–6): 375–381.

Appendices

1.1 Example of a Flatfile Header in ENA Format

```

ID      U54469; SV 1; linear; genomic DNA; STD; INV; 2881 BP.
XX
AC      U54469;
XX
DT      19-MAY-1996 (Rel. 47, Created)
DT      23-JUN-2017 (Rel. 133, Last updated, Version 5)
XX
DE      Drosophila melanogaster eukaryotic initiation factor 4E (eIF4E) gene,
DE      complete cds, alternatively spliced.
XX
KW      .
XX
OS      Drosophila melanogaster (fruit fly)
OC      Eukaryota; Metazoa; Ecdysozoa; Arthropoda; Hexapoda; Insecta; Pterygota;
OC      Neoptera; Holometabola; Diptera; Brachycera; Muscomorpha; Ephydroidea;
OC      Drosophilidae; Drosophila; Sophophora.
XX
RN      [1]
RP      1-2881
RX      DOI; .1074/jbc.271.27.16393.
RX      PUBMED; 8663200.
RA      Lavoie C.A., Lachance P.E., Sonenberg N., Lasko P.;
RT      "Alternatively spliced transcripts from the Drosophila eIF4E gene produce
RT      two different Cap-binding proteins";
RL      J Biol Chem 271(27):16393-16398(1996).
XX
RN      [2]
RP      1-2881
RA      Lasko P.F.;
RT      ;
RL      Submitted (09-APR-1996) to the INSDC.
RL      Paul F. Lasko, Biology, McGill University, 1205 Avenue Docteur Penfield,
RL      Montreal, QC H3A 1B1, Canada
XX
DR      MD5; 303680f06f3441eb47a0de3a028a8d06.
DR      FLYBASE; FBgn0015218; eIF-4E.

```

1.2 Example of a Flatfile Header in DDBJ/GenBank Format

```

LOCUS      DMU54469 2881 bp DNA linear INV 20-JUN-2017
DEFINITION Drosophila melanogaster eukaryotic initiation factor 4E (eIF4E)
            gene, complete cds, alternatively spliced.
ACCESSION  U54469
VERSION    U54469.1
KEYWORDS   .
SOURCE     Drosophila melanogaster (fruit fly)
  ORGANISM Drosophila melanogaster

```

Eukaryota; Metazoa; Ecdysozoa; Arthropoda; Hexapoda; Insecta; Pterygota; Neoptera; Holometabola; Diptera; Brachycera; Muscomorpha; Ephydroidea; Drosophilidae; Drosophila; Sophophora.

1 (bases 1 to 2881)

REFERENCE
AUTHORS Lavoie,C.A., Lachance,P.E., Sonenberg,N. and Lasko,P.
TITLE Alternatively spliced transcripts from the Drosophila eIF4E gene produce two different Cap-binding proteins
JOURNAL J. Biol. Chem. 271 (27), 16393-16398 (1996)
PUBMED 8663200

REFERENCE
AUTHORS Lasko,P.F.
TITLE Direct Submission
JOURNAL Submitted (09-APR-1996) Paul F. Lasko, Biology, McGill University, 1205 Avenue Docteur Penfield, Montreal, QC H3A 1B1, Canada

1.3 Example of a Feature Table in ENA Format

FH	Key	Location/Qualifiers
FH		
FT	source	1..2881
FT		/organism="Drosophila melanogaster"
FT		/chromosome="3"
FT		/map="67A8-B2"
FT		/mol_type="genomic DNA"
FT		/db_xref="taxon:7227"
FT	gene	80..2881
FT		/gene="eIF4E"
FT	mRNA	join(80..224,892..1458,1550..1920,1986..2085,2317..2404,2466..2881)
FT		/gene="eIF4E"
FT		/product="eukaryotic initiation factor 4E-I"
FT	mRNA	join(80..224,1550..1920,1986..2085,2317..2404,2466..2881)
FT		/gene="eIF4E"
FT		/product="eukaryotic initiation factor 4E-II"
FT	CDS	join(201..224,1550..1920,1986..2085,2317..2404,2466..2629)
FT		/codon_start=1
FT		/gene="eIF4E"
FT		/product="eukaryotic initiation factor 4E-II"
FT		/note="Method: conceptual translation with partial peptide sequencing"
FT		/db_xref="GOA:P48598"
FT		/db_xref="InterPro:IPR001040"
FT		/db_xref="InterPro:IPR019770"
FT		/db_xref="InterPro:IPR023398"
FT		/db_xref="PDB:4AXG"
FT		/db_xref="PDB:4UE8"
FT		/db_xref="PDB:4UE9"
FT		/db_xref="PDB:4UEA"
FT		/db_xref="PDB:4UEB"
FT		/db_xref="PDB:4UEC"
FT		/db_xref="PDB:5ABU"
FT		/db_xref="PDB:5ABV"
FT		/db_xref="PDB:5T47"
FT		/db_xref="PDB:5T48"
FT		/db_xref="UniProtKB/Swiss-Prot:P48598"
FT		/protein_id="AAC03524.1"
FT		/translation="MVVLETEKTSAPSTEQGRPEPPTSAAAPAEAKDVKPKEDPQETGE PAGNTATTTAPAGDDAVRTEHLYKHPLMNVWTLWYLENDRSKSWEDMQNEITSFDTVED FWSLYNHKPPSEIKLGSYSLFKKNIRPMWEDAANKQGGRWVITLNKSSKTDLDNLWL DVLLCLIGEAFDHSQICGAVINIRGKSNKISIWADGNNEEAALIEGHKLRDALRLGR NNSLQYQLHKDTMVKQGSNVKSIYTL"
FT	CDS	join(1402..1458,1550..1920,1986..2085,2317..2404,2466..2629)
FT		/codon_start=1
FT		/gene="eIF4E"

```

FT          /product="eukaryotic initiation factor 4E-I"
FT          /note="Method: conceptual translation with partial peptide
FT          sequencing; two alternatively spliced transcripts both
FT          encode 4E-I"
FT          /db_xref="GOA:P48598"
FT          /db_xref="InterPro:IPR001040"
FT          /db_xref="InterPro:IPR019770"
FT          /db_xref="InterPro:IPR023398"
FT          /db_xref="PDB:4AXG"
FT          /db_xref="PDB:4UE8"
FT          /db_xref="PDB:4UE9"
FT          /db_xref="PDB:4UEA"
FT          /db_xref="PDB:4UEB"
FT          /db_xref="PDB:4UEC"
FT          /db_xref="PDB:5ABU"
FT          /db_xref="PDB:5ABV"
FT          /db_xref="PDB:5T47"
FT          /db_xref="PDB:5T48"
FT          /db_xref="UniProtKB/Swiss-Prot:P48598"
FT          /protein_id="AAC03525.1"
FT          /translation="MQSDFHRMKNFANPKSMFKTSAPSTEQGRPEPPTSAAAPAEAKDV
FT          KPKEDPQETGEPAGNTATTTAPAGDDAVRTEHLYKHPLMNVWTLWYLENDRSKSWEDMQ
FT          NEITSFDTVEDFWSLYLNHIKPPSEIKLGS DYSLFKKNIRPMWEDAANKQGGRWVITLNK
FT          SSKTDLDLNWLVDVLLCLIGEAFDHSQICGAVINIRGKSNKISIWTDAGNNEEALEIG
FT          HKLRDALRLGRNNSLQYQLHKD TMVKQGSNVKSIYTL"

```

1.4 Example of a Feature Table in GenBank/DBJ Format

```

FEATURES             Location/Qualifiers
     source            1..2881
                     /organism="Drosophila melanogaster"
                     /mol_type="genomic DNA"
                     /db_xref="taxon:7227"
                     /chromosome="3"
                     /map="67A8-B2"
     gene              80..2881
                     /gene="eIF4E"
     mRNA              join(80..224,892..1458,1550..1920,1986..2085,2317..2404,
                     2466..2881)
                     /gene="eIF4E"
                     /product="eukaryotic initiation factor 4E-I"
     mRNA              join(80..224,1550..1920,1986..2085,2317..2404,2466..2881)
                     /gene="eIF4E"
                     /product="eukaryotic initiation factor 4E-II"
     CDS                join(201..224,1550..1920,1986..2085,2317..2404,2466..2629)
                     /gene="eIF4E"
                     /note="Method: conceptual translation with partial peptide
                     sequencing"
                     /codon_start=1
                     /product="eukaryotic initiation factor 4E-II"
                     /protein_id="AAC03524.1"
                     /translation="MVVLETEKTSAPSTEQGRPEPPTSAAAPAEAKDVKPKEDPQETG
                     EPAGNTATTTAPAGDDAVRTEHLYKHPLMNVWTLWYLENDRSKSWEDMQNEITSFDTV
                     EDFWSLYLNHIKPPSEIKLGS DYSLFKKNIRPMWEDAANKQGGRWVITLNKSSKTDLN
                     LWLDVLLCLIGEAFDHSQICGAVINIRGKSNKISIWTDAGNNEEALEIGHKLRDAL
                     RLGRRNNSLQYQLHKD TMVKQGSNVKSIYTL"
     CDS                join(1402..1458,1550..1920,1986..2085,2317..2404,
                     2466..2629)
                     /gene="eIF4E"
                     /note="Method: conceptual translation with partial peptide
                     sequencing; two alternatively spliced transcripts both
                     encode 4E-I"
                     /codon_start=1
                     /product="eukaryotic initiation factor 4E-I"

```

```

/protein_id="AAC03525.1"
/translation="MQSDFHRMKNFANPKSMFKTSAPSTEQGRPEPPTSAAAPAEAKD
VKPKEDPQETGEPAGNTATTTAPAGDDAVRTEHLYKHPMLMNWVTLWYLENDRSKSWED
MQNEITSFDTVEDFWSLYNHKPPSEIKLGS DYSLFKKNIRPMWEDAANKQGGRWVIT
LNKSSKTDLDNLWLDVLLCLIGFAFDHSDQICGAVINIRGKSNKISITWADGNNEEAA
LEIGHKLRDALRLGRNNSLQYQLHKDTMVKQGSNVKSIYTL"

```

6.1 Dynamic Programming

This appendix describes the basic dynamic programming method for RNA secondary structure prediction. An in-depth example is provided for prediction of a structure that uses a simplified energy model, suitable for manual calculation.

The dynamic programming algorithm is divided into two steps. In the first step of the algorithm, called the fill step, the lowest conformational free energy possible for each sub-fragment of sequence, from nucleotide i to nucleotide j , is calculated. Two arrays are stored, $V(i,j)$ and $W(i,j)$. In $V(i,j)$, the lowest free energy from nucleotide i to nucleotide j , inclusive, is stored with the requirement that i base pair to j . Similarly, $W(i,j)$ is the lowest free energy for the same nucleotide fragment, requiring that $W(i,j)$ will be incorporated into a multibranch loop and there be at least one base pair. One way to fill the arrays is to start with the shortest sequence fragments that can fold into a structure (five nucleotides) and then fill the arrays for increasingly longer fragments (6-mers, 7-mers, 8-mers, and so forth).

The $V(i,j)$ and $W(i,j)$ arrays are filled, taking advantage of recursion. For example, the free energy change of a base pair i and j , stacked on paired nucleotides $i + 1$ and $j - 1$, is:

$$\Delta G_{37}^{\circ} = V(i + 1, j - 1) + \Delta G_{37}^{\circ} \text{ (base pair stacking)} \quad (6.A.1)$$

where $V(i + 1, j - 1)$ was calculated previously because it corresponds to a shorter sequence fragment. Once the arrays are filled, the lowest conformational free energy possible for the sequence is determined, but the structure is not yet known. The second step of the dynamic programming algorithm, called traceback, determines the secondary structure that is the lowest free energy conformation.

As an illustration of dynamic programming for secondary structure prediction, consider the folding of rGCGGGUACCGAUCGUCGC for which the number of hydrogen bonds in canonical pairs will be maximized. This calculation is simpler than free energy minimization, but it illustrates the important points of dynamic programming. The following recursions will be used for $1 \leq i < j \leq N$, where N is the number of nucleotides in the sequence:

$$\begin{aligned} V(i,j) &= 0 \text{ if } i \text{ and } j \text{ cannot pair canonically} \\ &= \max[V_{\text{hairpin}}[i,j], V_{\text{stack/internal/bulge}}[i,j], V_{\text{multibranch}}[i,j]] \text{ if } i \text{ and } j \text{ can pair} \end{aligned} \quad (6.A.2)$$

$$W(i,j) = \max[V[i,j], W[i + 1,j], W[i,j - 1], W[i,k] + W[k + 1,j] \text{ for } i < k < j] \quad (6.A.3)$$

$V(i,j)$ is the maximum number of hydrogen bonds for the sequence fragment from nucleotides i to j with i and j paired. So, as shown in Eq. (6.A.2), to fill $V(i,j)$ the possibilities that the pair of i and j can close a hairpin loop, stack on a previous pair, close an internal loop, close a bulge loop, or close a multibranch loop all need consideration. This accounts for any type of structure that the i to j base pair can close. Each term counts the total hydrogen bonds:

$$\begin{aligned} V_{\text{hairpin}} &= \text{number of hydrogen bonds in pair } i \text{ and } j, \text{ if } j - i > 3 \\ &= 0, \text{ if } j - i \leq 3 \end{aligned} \quad (6.A.4)$$

$$\begin{aligned} V_{\text{stack/internal/bulge}} &= (\text{number of hydrogen bonds in pair } i \text{ and } j) \\ &\quad + \max[V(k_1, k_2) \text{ for } i < k_1 < k_2 < j] \end{aligned} \quad (6.A.5)$$

$$\begin{aligned} V_{\text{multibranch}} &= (\text{number of hydrogen bonds in pair } i \text{ and } j) \\ &\quad + \max[W(i + 1, k) + W(k + 1, j - 1) \text{ for } i + 1 < k < j - 1] \end{aligned} \quad (6.A.6)$$

$V(i,j)$ is zero when i and j cannot form a canonical pair. Furthermore, hairpin loops that enclose fewer than three unpaired nucleotides are forbidden by assigning those loops 0 hydrogen bonds (Eq. (6.A.4)). AU and GU pairs have two hydrogen bonds and a GC pair has three hydrogen bonds. $W(i,j)$ is the maximum number of hydrogen bonds for the fragment of nucleotides i to j , without the constraint that i and j must be paired. The term $W(i,k) + W(k+1,j)$ in Eq. (6.A.3) allows any number of helical branches in a multibranch loop by recursion. In other words, $W(i,k)$, for example, could have previously been composed of two branches. $V(i,j)$ and $W(i,j)$ are filled by considering all 5-mers, then 6-mers, then 7-mers, etc. of sequence length. Note that $V(i,j)$ and $W(i,j)$ for sequences shorter than five nucleotides are zero because of the assumption in Eq. (6.A.4) that the minimum length of a hairpin is three unpaired nucleotides. Figure 6.A.1 shows the pseudo-computer code for the fill order.

Figures 6.A.2 and 6.A.3 show the filled $V(i,j)$ and $W(i,j)$ arrays, respectively. The values of some entries are instructive and the conformations with the maximum hydrogen bonds for those entries are illustrated in Figure 6.A.4. Consider, for example, $V(14,18) = 3$. This is the number of hydrogen bonds in the pair of G14 and C18, with G14 and C18 closing a hairpin loop. $V(10,16) = 5$ is a stack of G10 and C16 onto the pair of A11 and U15. $V(6,14) = 4$ is the pair of U6 and G14 closing a bulge loop, for which a pair between A7 and U12 close the interior end of the loop. Finally, $V(2,17) = 14$ is the closure of a multibranch loop by C2 and G17 with the bifurcation of branches represented by $W(3,9)$ and $W(10,16)$. $W(14,18) = 3$ is equal to $V(14,18)$. $W(8,18) = 8$ is the extension of an unpaired nucleotide on $W(9,18)$ (which

```

L = 5
i = 1
While (L ≤ N) {
  j = i + L-1
  Calculate V(i,j) according to equation 6.A.2
  Calculate W(i,j) according to equation 6.A.3
  If (j < N) i = i + 1
  Else {
    i = 1
    L = L + 1
  }
}

```

Figure 6.A.1 Pseudo-computer code for the fill order of $V(i,j)$ and $W(i,j)$. This is one representative scheme for the direction of filling of the two-dimensional arrays. This scheme calculates $V(i,j)$ and $W(i,j)$ for each 5-mer, then 6-mer, then 7-mer, etc., starting from the 5' end of the sequence.

V	i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
j		G	C	G	G	G	U	A	C	C	G	A	U	C	G	U	C	G	C
18	C	17	0	14	11	11	0	0	0	0	6	0	0	0	3	0	0	0	
17	G	0	14	0	0	0	7	0	8	8	0	0	2	3	0	0	0		
16	C	14	0	10	9	8	0	0	0	0	5	0	0	0	0	0			
15	U	13	0	9	7	6	0	5	0	0	2	2	0	0	0				
14	G	0	11	0	0	0	4	0	3	3	0	0	0	0					
13	C	12	0	8	7	5	0	0	0	0	0	0	0						
12	U	11	0	5	5	4	0	2	0	0	0	0							
11	A	0	0	0	0	0	2	0	0	0	0								
10	G	0	9	0	0	0	2	0	0	0									
9	C	6	0	6	3	3	0	0	0										
8	C	3	0	3	3	0	0	0											
7	A	0	0	0	0	0	0												
6	U	2	0	0	0	0													
5	G	0	0	0	0														
4	G	0	0	0															
3	G	0	0																
2	C	0																	
1	G																		

Figure 6.A.2 The filled $V(i,j)$ array for sequence GCGGGUACCGAUCGUCGC.

W	i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
j		G	C	G	G	G	U	A	C	C	G	A	U	C	G	U	C	G	C
18	C	17	14	14	11	11	8	8	8	8	6	3	3	3	3	0	0	0	
17	G	14	14	11	11	8	8	8	8	5	3	3	3	0	0	0			
16	C	14	11	11	9	8	5	5	5	5	2	0	0	0	0				
15	U	13	11	9	7	6	5	5	3	3	2	2	0	0	0				
14	G	12	11	8	7	5	4	3	3	3	0	0	0	0					
13	C	12	9	8	7	5	2	2	0	0	0	0	0						
12	U	11	9	6	5	4	2	2	0	0	0	0							
11	A	9	9	6	3	3	2	0	0	0	0								
10	G	9	9	6	3	3	2	0	0	0									
9	C	6	6	6	3	3	0	0	0										
8	C	3	3	3	3	0	0	0											
7	A	2	0	0	0	0	0												
6	U	2	0	0	0	0													
5	G	0	0	0	0														
4	G	0	0	0															
3	G	0	0																
2	C	0																	
1	G																		

Figure 6.A.3 The filled $W(i,j)$ array for sequence GCGGGUACCGAUCGUCGC.

Array:	Representation:
$V(14,18) = 3$	
$V(10,16) = 5$	
$V(6,14) = 4$	
$V(2,17) = 14$	
$W(8,18) = 8$	
$W(4,17) = 11$	

Figure 6.A.4 Illustrations of maximum hydrogen bond conformations as found by the recursions. Regions in gray are not required by recursions and the conformation in those regions is unknown as the arrays are filled recursively. For example, $V(10,16) = 5$ is a case in which a base pair stacks on a previous pair, nucleotide 11 base paired to nucleotide 15. The recursions utilize $V(11,15)$, but the structure for the region between nucleotides 11 and 15 is unknown as $V(10,16)$ was previously calculated and is therefore drawn in gray.

is equal to the extension of an unpaired nucleotide on $W(8,17)$). $W(4,17) = 11$ is a bifurcation, i.e. the optimal structure from nucleotides 4 to 17 has more than one branch, and is the sum of $W(4,8)$ and $W(9,17)$.

At this point, it is clear that the maximum number of hydrogen bonds for this sequence is 17 and is represented by $V(1,N)$. To find the structure that has 17 hydrogen bonds, the traceback must be performed as illustrated in Figure 6.A.5. Traceback starts by placing 1, N , and the maximum number of hydrogen bonds on the stack. The stack is a storage device that can expand to accommodate as many sequence fragments as needed for the structure traceback. At each step of the main loop, a triple, consisting of i , j , and the number of hydrogen bonds, is taken from the stack. For this sequence, in the first step through the loop, $V(1,18)$ equals 17,

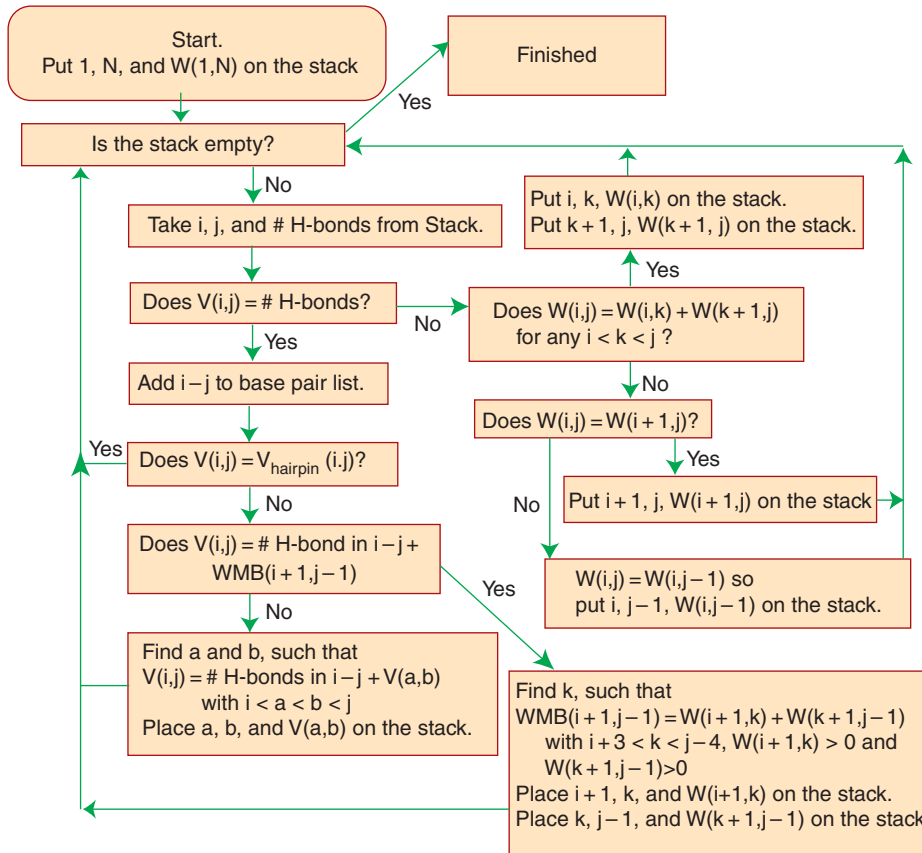


Figure 6.A.5 Flowchart for structure traceback. Traceback starts by placing 1, N , and the maximum number of hydrogen bonds for the sequence onto the stack and proceeds until the stack is empty.

the number of hydrogen bonds. Therefore, nucleotides 1 and 18 are base paired. Following the flowchart, $V(1,18)$ is not equal to the hairpin term and there is no bifurcation such that $V(1,18) = W(1,k) + W(k+1,18)$. Instead, $V(1,18) = V(2,17) + 3$ (the number of hydrogen bonds in the GC pair), so 2, 17, and $V(2,17) = 14$ are placed on the stack.

When taking $V(2,17)$ from the stack, again $V(2,17) = 14$, the number of hydrogen bonds, so nucleotide 2 is base paired to 17. $V(2,17)$ does not equal $V_{\text{hairpin}}(2,17)$. $V(2,17)$ does equal $W(3,9) + W(10,16)$, so 3, 9, and $W(3,9) = 6$ are placed on the stack, as are 10, 16, and $W(10,16) = 5$.

Now, 10, 16, and 5 are taken from the stack. Again, $V(10,16) = 5$, so nucleotide 10 is paired to 16. Following down the flowchart, $V(10,16)$ is found to be $V(11,15) + 3$, the number of hydrogen bonds in the base pair of G10 and C16. Therefore, 11, 15, and $V(11,15) = 2$ are placed on the stack. Next 11, 15, and 2 are removed from the stack, A11 and U15 are paired, and $V(11,15) = V_{\text{hairpin}}(11,15)$, so nothing is placed on the stack as this branch has been followed to its termination in a hairpin loop.

Now, 3, 9, and 6 are taken from the stack. This branch is similar to the branch starting with nucleotides 10 and 16, in that it has two base pairs: G3–C9 and G4–C8. After finding those two pairs, the stack is empty and the structure with 17 hydrogen bonds, as illustrated in Figure 6.A.6, is determined.

The scaling of RNA secondary structure is limited by multibranch loop searching, which must be done for every sub-fragment in the sequence, i.e. from nucleotides i to j with $i < j \leq N$. The search looks for a bifurcation for which the fragment is divided by k into two segments with stems, so $i < k < j$. Therefore, three indices are being

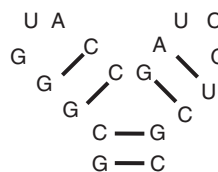


Figure 6.A.6 The secondary structure of rGCGGGUACCGAUCGUCGC with 17 hydrogen bonds.

searched over N nucleotides in the worst case step, the fragment that stretches the length of the sequence, from 1 to N . This is N^3 steps. Internal loop searching, which would naturally require a search over $i < i' < j' < j$, and therefore be $O(N^4)$, can be limited in two ways. The traditional method is to simply limit the size of internal loops so that $i' - i + j - j' < M$, where M is the maximum loop size. A reasonable limit in practice is $M = 30$. A second method has also been devised that does not limit the size of internal loops, but instead takes advantage of the form of the internal loop nearest neighbor parameters to pre-fill the arrays, splitting the N^4 process into two N^3 steps (Lyngsø et al. 1999).

Reference

Lyngsø, R., Zuker, M. & Pederson, C. (1999). Fast evaluation of internal loops in RNA secondary structure prediction. *Bioinformatics*. 15, 440–445.

Glossary

- 16S ribosomal RNA gene** A gene, found in all prokaryotes, that encodes a key RNA component of the ribosome. This gene is often present in multiple copies in a genome.
- 2D gel electrophoresis** Two-dimensional polyacrylamide electrophoresis. A technique for separating large numbers of proteins by loading them onto a gel, applying an electric current to separate by isoelectric point (pI), and then, in the perpendicular direction, separating by molecular weight.
- ab initio** Latin for “from the beginning.” A term frequently used in chemistry and physics to indicate that the method uses no prior knowledge or that it uses basic truths or fundamental principles to determine a given result or structure.
- accessible surface area (ASA)** The surface area of a protein or macromolecule that could be contacted by water molecules or other solvent/solute molecules, measured in square ångströms (Å²). Often used in assessing the quality of protein structures and the strength of hydrophobic interactions.
- admixture** Mating between individuals from two different populations. Admixture between two populations is typically measured by a rate, varying from 0 to 1.
- alert** A decision support rule or “pop-up” in an electronic health record system, based on a rule.
- algorithm** Any sequence of actions (such as computational steps) that are used to perform a particular task.
- alignment** Two or more sequences that have been lined up, matching as many identical residues or conservatively substituted positions as possible.
- allele** Any of the forms of a gene that may occur at a given locus. In simple Mendelian inheritance, dominant alleles are expressed more than recessive alleles.
- allele frequency spectrum** A graph in which allele frequency bins are plotted on the *x*-axis and the number (or proportion) of occurrences of each allele frequency is plotted on the *y*-axis.
- alpha diversity** The distribution of taxa (often expressed as number and evenness) in a given sample.
- alpha parameter** A value that affects the shape of a gamma distribution. In phylogenetic analyses, describes whether the rate of change across sites is high or low. The alpha parameter can result in gamma curves ranging from bell shaped ($\alpha > 1$) to L shaped ($\alpha < 1$).
- alternative splicing** The process through which a cell can generate many different protein products from a single gene. A gene is transcribed into a primary RNA transcript, containing both exons and introns; these exons (and sometimes introns) can then be combined into one or more different messenger RNA (mRNA) molecules, each encoding for a different protein.
- analogous** In phylogenetics, characters that have descended in a convergent fashion from unrelated ancestors.
- analysis of variance (ANOVA)** A statistical method used to test differences between the means (or averages) of two or more groups. ANOVA is more commonly used when analyzing three or more groups, while Student’s *t*-tests are used to analyze two groups only.

- ancestry** The origin, typically geographic, of a person's ancestors.
- annotated spectrum library** A curated, non-redundant collection of annotated peptide spectra used for the identification of proteins through spectral library search (matching) algorithms.
- application programming interface (API)** Definitions, protocols, and tools used to build software applications, including specifications for communications between systems.
- ASCII** Acronym for *American Standard Code for Information Interchange*. ASCII codes represent the text characters that can be typed on a conventional computer or typewriter keyboard. When someone asks for a file in ASCII format, it means they want plain text with no formatting such as tabs, bold, or underscoring. ASCII is the raw text format that any computer can understand.
- ASN.1** Acronym for *Abstract Syntax Notation One*. ASN.1 is a formal language for abstractly describing messages or information to be exchanged. ASN.1 is used extensively by the National Center for Biotechnology Information (NCBI) in representing sequence, structure, interaction, mapping, and bibliographic records.
- artificial neural network** A machine learning or artificial intelligence technique that learns patterns through repeated rounds of training. Neural networks are mathematical constructs designed to mimic the brain with a series of interconnected nodes that grow stronger or weaker with training.
- attachment site** A short sequence (or pair of sequences) that are used by bacteriophages to insert into bacterial chromosomes. Bacteriophage insertion (also known as *lysogeny*) requires a phage-encoded integrase that promotes reciprocal strand exchange between phage and bacterial attachment sites. Most bacterial attachment sites are 15–25 base pairs long.
- average mass** The mass of an ion calculated by averaging all common isotope variations. This quantity is typically used when the resolution of the instrumentation is not adequate to distinguish individual isotopes.
- backbone** This refers to the collection of common atoms or angles that constitute the heavy atoms (C, N, O) that make up every amino acid residue and peptide. Backbone atoms are equivalent to the four heavy atoms (N, C α , C, and O) found in glycine residues and therefore exclude the side-chain atoms that make each biogenic amino acid unique. Backbone atoms and backbone angles define the general shape of the polypeptide, including its secondary and tertiary structure.
- balancing selection** A form of natural selection in which multiple alleles at a locus are maintained at intermediate frequencies, often because the heterozygous genotype has a selective advantage.
- base** A chemical ring structure that is part of a nucleotide. Bases form hydrogen bonds with other bases to form base pairs.
- base caller** A program used to convert raw sequencer output to an ordered list of base identities and quality scores.
- Bayesian network** A method in machine learning used to predict a feature in a dataset given some known but incomplete information. Based on Bayes' rule for conditional probability.
- beta diversity** The dissimilarity between two samples, often expressed in taxonomic terms.
- BLOSUM matrix** BLOCKS substitution matrix. Also see *PAM matrix*.
- Bonferroni correction** A conservative statistical multiple comparison correction used to correct the significance threshold (alpha value) of multiple independent statistical tests, which together may have a greater chance of generating false positives than the individual test.
- Boolean** Refers to an expression or variable that can have only a true or false value. Named after George Boole, a British mathematician who developed the theory of algebraic logic or Boolean algebra, which is now used in almost all electronic computation.

- bootstrapping** The process of randomizing and sampling positions in an alignment in order to determine the degree of support for a given node within a phylogenetic tree. Bootstrap values are expressed as a percentage, with higher percentages reflecting the number of times a particular branching order is recovered and indicating that a node is well supported.
- bottom-up proteomics** Another term for shotgun proteomics, referring to the fact that the analysis begins with the peptide constituents of a sample. Also see *shotgun proteomics*.
- browser** Program used to access sites on the World Wide Web. Using hypertext markup language (HTML), browsers are capable of representing a web page the same way regardless of computer platform. Also see *hypertext markup language*.
- candidate gene** A gene that is implicated in the causation of a disease. The protein product of a candidate gene may implicate the candidate gene as being the actual disease gene being sought.
- cDNA library** A collection of double-stranded DNA sequences that are generated by copying mRNA molecules. Since these sequences are derived from mRNAs, they contain only protein-coding DNA.
- centimorgan (cM)** The genetic distance between two markers that recombine at a frequency of 1%. In humans, 1 cM is approximately equivalent to one million base pairs.
- centromere** A specialized DNA sequence located near the center of a eukaryotic chromosome that links a pair of sister chromatids. Centromeres contain long stretches of highly repetitive DNA and serve as a site of assembly for the kinetochore, the master regulator of chromosome segregation.
- characters and character states** In phylogenetics, characters are homologous features in different organisms. The exact condition of that feature in a particular individual is the character state. As an example, the character “hair color” can have the character states “gold,” “red,” and “yellow.” In molecular biology, the character states can be one of the four nucleotides (A, C, T, G) or one of the 20 amino acids. Please note that some authors define character to mean the character state as defined here.
- chromatin immunoprecipitation (ChIP)** An experimental method that is used to deduce how proteins interact with specific kinds of DNA binding sites, such as transcription factors, promoters, or other upstream regulatory elements.
- chromatogram** A file containing raw data and ancillary information about a single DNA sample that has been run through an automated DNA sequencing instrument. Also see *base caller*.
- clade** A monophyletic taxon. A group of organisms or genes that includes the most recent common ancestor of all its members and all of the descendants of that most recent common ancestor. From the Greek *klados*, meaning branch.
- clade annotation** An approach to genome and proteome annotation that is based on the observation that gene and protein functions and sequences are not globally conserved across all species but are often locally conserved in separate clades. Clade annotation assists in getting around the issue of weak sequence conservation in consensus signals.
- cladistics** A method for hypothesizing relationships among organisms, genes, or proteins based on shared characteristics.
- cladogram** A branching diagram showing the relationships between clades.
- client** A computer, or the software running on a computer, that interacts with another computer at a remote site (server). Note the difference between client and user.
- coalescence** The principle that population genetic variation at a locus can ultimately be traced back to a single ancestor.
- coding bias** Also *codon usage bias*. This refers to differences in the frequency of occurrence of synonymous codons in protein-coding DNA. The frequency of certain codons in protein-coding DNA (as opposed to non-coding DNA) and in certain organisms is often biased toward a small set of possible codons for the same amino acid. Codon bias reflects a balance between mutational biases and natural selection for translational optimization.

- coding sequence (CDS)** A segment of genomic DNA or cDNA that codes for a protein. This abbreviation is used extensively in sequence database records. Also see *complementary DNA*.
- coding statistics** A mathematical function that computes a real number related to the likelihood that a given DNA sequence codes for a protein. Coding statistics provide information about codon bias, third-base wobble, the frequency of certain hexamers, the length of coding regions, and similar types of DNA content measures.
- codon** The triplet of bases in either a DNA or RNA sequence that ultimately codes for a specific, single amino acid.
- combinatorics** A branch of mathematical science that is concerned with the calculation or enumeration of different combinations of objects or states. It has many applications ranging from physics and probability theory to biology and computer science.
- comparative proteomics** The general approach of comparing proteomes from two or more cellular states, then using mass spectrometry to identify the proteins or peptides that differ between them.
- comparative sequence analysis** A method used to determine RNA secondary structure in which a sequence alignment is used to infer locations of base pairs.
- complementary** Two sequences that can form an uninterrupted helix of Watson–Crick base pairs.
- complementary DNA (cDNA)** Single-stranded DNA that has been synthesized from an mRNA template by reverse transcriptase.
- consensus** In alignments, the base or amino acid most likely to occur at any given position; consensus sequences can be used to characterize protein families.
- conservative substitution** The replacement of one residue by another residue having similar properties such as size, charge, and hydrophobicity.
- contig** Short for *contiguous*. Refers to a contiguous set of overlapping DNA sequences.
- copy number variant** Population variation in the number of copies (per individual) of a series of consecutive DNA base pairs, typically defined as 50 or more base pairs.
- copy number variation** A gain or loss in genetic material resulting from deletions or duplications that may be associated with a specific disease state.
- core genome** The set of genes present in all individuals in a clade.
- correlation coefficient** A numerical measure of the strength and direction of a linear relationship between two variables; also a quantitative method for evaluating the correlation or statistical relationship between two types of measurements, observables, or variables. Correlation coefficients range between -1 and $+1$, where $+1$ indicates the strongest possible agreement and -1 indicates the strongest possible disagreement. The Pearson correlation coefficient (r or R) is the most commonly used correlation coefficient and is defined as the covariance of the variables divided by the product of their standard deviations. The correlation coefficient is often mistaken for the less useful coefficient of determination (r^2 or R^2), which is the square of the correlation coefficient.
- cytogenetic map** The representation of a chromosome upon staining and examination by microscopy. Visually distinct light and dark bands give each chromosome a unique morphological appearance and allow for the visual tracking of cytogenetic abnormalities such as deletions or inversions.
- data-dependent acquisition (DDA)** Mode of data collection in tandem mass spectrometry in which randomly selected precursor ions within a specified mass range are subjected to molecular fragmentation.
- data-independent acquisition (DIA)** Mode of data collection by a mass spectrometer in which all precursor ions within a specified mass range are subjected simultaneously to fragmentation.
- data exchange format** A structured, standard way of encoding or writing down data for the purpose of data exchange between groups of people (or computer systems) that have agreed upon the format.

- dalton (Da)** A unit of molecular mass equal to the mass of a hydrogen atom. On average, amino acids are ~110 Da, while DNA/RNA bases are ~330 Da.
- de novo sequencing** Technique to derive the peptide sequence from a tandem mass spectrometry (MS²) spectrum without the use of a sequence database.
- deconvolution** General term referring to the mathematical process of disentangling multiple components in a spectrum to identify the base constituents. In mass spectrometry, it refers to reducing multiple charge-state peaks into a single cognate mass peak.
- deisotoping** Removal of companion isotope peaks in a convoluted spectrum, to represent the fundamental ion species as a single data point. Deisotoping is commonly performed to reduce data complexity, usually in conjunction with charge state deconvolution.
- deletion** A mutation in which one or more bases is lost from a given region of a chromosome.
- deletion/insertion polymorphism (DIP)** Alleles that are represented by one base or more that are present in one sequence and absent in the other.
- descriptor** Information about a sequence or set of sequences whose scope depends upon its placement in a record. Placed on a set of sequences to reduce the need to save multiple redundant copies of information.
- difference gel electrophoresis (DIGE)** A type of two-dimensional polyacrylamide gel electrophoresis in which two samples are run on a single gel.
- dihedral angle** Also torsion angle. The angle between two intersecting planes. In chemistry, it is the angle between planes through two sets of three atoms, having two atoms in common. Main chain dihedral angles are labeled as ϕ , ψ , and ω (where ω corresponds to the peptide backbone), while side-chain dihedral angles are labeled as χ_1 , χ_2 , and so forth.
- dN/dS ratio** The ratio of synonymous vs. non-synonymous substitutions that is used to determine whether sequences are undergoing positive or negative selection. If $dN/dS \sim 1$, no selection has occurred. If $dN/dS < 1$, negative (or purifying) selection has occurred, removing alleles that are deleterious. If $dN/dS > 1$, positive selection has occurred, favoring alleles that are advantageous.
- domain name** Refers to one of the levels of organization of the internet and used to both classify and identify host machines. Top-level domain names usually indicate the type of site or the country in which the host is located.
- dotplot** A visual technique for comparing two sequences with one another, allowing for the identification of regions of local alignment, direct or inverted repeats, insertions, deletions, or low-complexity regions.
- download** The act of transferring a file from a remote host to a local machine via FTP. Also see *file transfer protocol*.
- dynamic programming** A computational technique used to solve complex problems by decomposing the problem into successively smaller subproblems, then solving them recursively. The solution of a subproblem of given complexity is dependent on the solutions already computed for subproblems of lesser complexity.
- eigenvector** A non-zero vector of which all values change by the same scalar factor when that transformation is applied to it.
- electron impact (or ionization) mass spectrometry (EI-MS)** The technique employed by gas chromatography mass spectrometers (GC-MS). It was one of the first ionization techniques developed and uses high-energy electrons fired at a specific energy (70 eV) to fragment and ionize molecules for mass analysis. EI-MS is known as a hard ionization method, as most molecules are “shattered” to tiny bits. Every molecule breaks apart in a characteristic pattern that serves as a kind of EI-MS fingerprint.
- electronic polymerase chain reaction (e-PCR)** A computational method that predicts the location of sequence tagged sites (STSs) in DNA by searching for sub-sequences that closely match the PCR primers used to make the STS; these sub-sequences must also have the correct order, orientation, and spacing such that they could prime the amplification of a PCR product of the correct molecular weight. Also see *sequence tagged sites*.

electrospray ionization (ESI) A method for creating ions in mass spectrometry that consists of forming a spray of charged droplets containing analyte molecules, then de-solvating the droplets, leaving charged ions for analysis.

electrospray ionization tandem mass spectrometry (ESI-MS/MS) Electrospray ionization is a soft ionization method that keeps the parent molecules mostly intact. It works by spraying the molecules through an electrified nozzle, which gives the molecules a charge. When the ionized molecules are sent into the tandem mass spectrometer, they collide with inert gas molecules and are gently broken apart into smaller fragment ions. Every molecule breaks apart in a characteristic pattern that serves as a kind of ESI-MS/MS fingerprint. Also see *tandem mass spectrometry*.

energy minimization A computational method for reducing the calculated covalent and non-covalent energy of a molecule with a given geometry. Energy minimization uses highly parameterized Newtonian descriptions of molecular bonds and atomic interactions. Energy minimization is frequently used to refine or fix protein structures determined from X-ray, nuclear magnetic resonance (NMR), or homology modeling.

enhancer A short (50–1500 bp) region of DNA that can be preferentially bound by transcription factor proteins to increase the likelihood that transcription of a nearby gene will occur.

expressed sequence tags (ESTs) Short (300–500 bp) single reads from mRNA (cDNA) which are usually produced in large numbers. They represent a snapshot of what is expressed in a given tissue or at a given developmental stage. They represent tags (some coding, others not) of expression for a given cDNA library. Also see *cDNA library*.

exon The part of a gene that remains in a mature mRNA transcript after any introns have been spliced out; the “expressed region” of a gene.

extensible markup language (XML) A text markup language for interchanging structured data that play an increasingly important role in the exchange of a wide variety of data on the web and elsewhere.

family trio DNA samples from mother, father, and child.

feature Annotation on a specific location on a given sequence.

Fast Healthcare Interoperability Resources (FHIR) An API standard for accessing electronic health record systems. Also see *application programming interface*.

feature A variable or data field used in predictive analytics for the purposes of clustering or supervised machine learning approaches.

file transfer protocol (FTP) The method by which files are transferred between hosts.

filtering See *masking*.

firewall A computer separating a company or organization’s internal network from the public part, if any, of the same network. Intended to prevent unauthorized access to private computer systems.

flanking sequence Sequences 5’ or 3’ of a core sequence of interest.

Fourier transform mass spectrometry (FT/MS) Also Fourier transform ion cyclotron resonance MS (FTICR). A highly accurate ion measurement used especially for large molecules of up to 1 million daltons or more. Electromagnetic forces are used to cycle ions in a chamber, which are then measured by the frequency at which they resonate. To convert from frequency to a mass-charge (m/z) spectrum, a Fourier transform is applied. Also see *mass to charge ratio*.

fragmentation The formation of fragment ions arising from the cleavage of a protein or peptide backbone (or side chains) due to dissociation of energetically unstable molecular ion states.

frameshift mutation Mutations arising from insertion or deletion of nucleotides whose length is not evenly divisible by 3; this leads to a change in reading frame, meaning that protein translation from the point of the mutation onward will be incorrect.

- functional profiling** Prediction of functions, pathways, and metabolic modules in the microbiome by comparing predicted genes from a metagenome sample with a reference database such as the Kyoto Encyclopedia of Genes and Genomes (KEGG) or Swiss-Prot.
- gamma distribution** A probability distribution that describes the degree of rate substitution heterogeneity (the degree of variability of change) across sites.
- gap** Used to improve alignments between sequences. Gaps theoretically represent insertions and deletions between sequences being studied.
- gene flow** The exchange of DNA between two different populations.
- gene ontology (GO)** A formal system for describing gene function. GO is composed of two parts, an ontology and a set of annotations. The ontology defines terms that describe gene function that are organized in a hierarchy from least specific terms at the top to most specific at the bottom. The ontology covers three aspects of gene function: biological process (e.g. tricarboxylic acid cycle), cellular component (e.g. cytoplasm), and molecular function (e.g. kinase activity). Annotations link ontology terms to genes, along with a description of the evidence used to make the link. Each gene can be linked to multiple annotations.
- gene set** Simply, a set of related genes. A *pathway gene set* includes all genes in a pathway. Gene sets can be based on various relationships between genes, such as cellular localization (e.g. nuclear genes) or enzymatic function (e.g. protein kinases). Details such as protein interactions are not included.
- genetic drift** Random variation in population allele frequencies through time, arising as a result of finite population size.
- genetic interaction** An interaction defined by a change in phenotype caused by a change in genotype. For instance, if two genes that are genetically altered (e.g. knocked out) do not lead to a change in phenotype on their own but, when altered together, cause the death of the organism, then the two genes have a synthetic lethal interaction. This may indicate that the genes are part of parallel pathways that impinge upon an essential process.
- genetic map** Gives the relative positions of known genes and/or markers. Markers must have two or more alleles that can be readily distinguished.
- genetic marker** A DNA feature whose physical location is known and that can be used to (indirectly) deduce the mode of inheritance of a gene.
- genome** All of the DNA found within each of the cells of an organism. Eukaryotic genomes can be subdivided into their nuclear genome (chromosomes found within the nucleus) and their mitochondrial genome. Plants also contain chloroplast genomes.
- genome-wide association study** A method through which genetic markers from populations of both affected and unaffected individuals are systematically scanned in order to identify specific mutations or genetic variations associated with a given disease.
- genotype** [1] The alleles present in a given individual's DNA for a particular genetic marker or set of markers. [2] The unique genetic makeup of an organism. Also see *phenotype*.
- graph** In phylogenetics, a set of vertices (also called nodes) and a set of edges connecting those vertices. Can be visualized as a set of points connected by lines.
- graph theory** A field of computer science and mathematics that focuses on the development of algorithms and proofs concerning graphs.
- graphical user interface (GUI)** Refers to software front-ends that rely on pictures and icons to direct the interaction of users with the application.
- haplotype** Sets of alleles that are usually inherited together. The haploid sequence inherited from each parent.
- heat map** A two-dimensional representation of data (such as expression data) in which data are represented as a series of colors.
- heuristic algorithm** An economical strategy for deriving a solution to a problem that is not guaranteed to find the optimal solution. Required when the algorithm for the optimal solution is computationally impractical.

- hexamer frequency** The frequency of a given six-nucleotide segment compared with the frequency of the same hexamer as derived from the general base composition of the entire genome. The frequencies of certain hexamers can differ greatly between coding DNA and non-coding DNA.
- hidden Markov model (HMM)** A probabilistic method for the linear analysis of sequences. HMMs are used in bioinformatics for almost any task that can be described as a process that analyzes sequences from left to right. Applications of HMMs to biological data include gene prediction, protein secondary structure prediction, and the detection of sequence signals such as translation initiation sites.
- homologs** In phylogenetics, particular features in different individuals that are genetically descended from the same feature in a common ancestor are termed homologous. Also see *orthologs* and *paralogs*.
- homologous (from homolog)** Homologous sequences are sequences that serve a similar function or are genetically related. Homologous nucleotides are nucleotides that serve the same function in different sequences.
- homology modeling** Also comparative modeling. A method for predicting the tertiary structure of a protein by using an existing homologous protein structure as a template.
- homoplasy** Similarity that has evolved independently and is not indicative of common phylogenetic origin.
- horizontal gene transfer** The movement of genetic material from one organism to another, including by transformation, transduction, or conjugation. Also see *vertical gene transfer*.
- host** Any computer on the internet that can be addressed directly through a unique IP address.
- HPO** Human Phenotype Ontology.
- hypertext** Within a web page, text that is differentiated either by color or by underlining which functions as a hyperlink. Also see *uniform resource locator*.
- hypertext markup language (HTML)** The standard, text-based language used to specify the format of World Wide Web documents. HTML files are translated and rendered through the use of web browsers.
- hyperlink** A graphic or text within a web document that can be selected using a mouse. Clicking on a hyperlink transports the user to another part of the same web page or to another web page, regardless of location.
- identity** A quantitative measure of how related two sequences are to one another, assessed as the total number of exact matches in a pairwise sequence alignment.
- InChI key** A fixed length (27 character) condensed digital representation of an InChI that is not human understandable, developed in order to facilitate web searches for chemical compounds as many full length InChI identifiers are too long for search engines to handle. Also see *IUPAC International Chemical Identifier*.
- indel** Acronym for *insertion or deletion*. Applied to length-variable regions of a multiple alignment when it is not specified whether sequence length differences have been created by insertions or deletions.
- insertion** A mutation in which one or more bases are inserted into a chromosomal region.
- insulator** A long-range genetic regulatory element found in eukaryotes that blocks the interaction between enhancers and promoters. Insulators contain clustered binding sites for sequence-specific DNA binding proteins and mediate intra- and interchromosomal interactions.
- integrase** A viral enzyme that catalyzes the integration of virally derived DNA into the host cell DNA in the nucleus, forming a provirus that can be activated to produce viral proteins.
- interaction** Any relationship, physical or otherwise, between biological entities such as proteins, cells, and amino acids that can be defined experimentally.
- interactome** The set of all interactions in an organism.
- intergenic region** A stretch of non-coding DNA located between genes. Some intergenic DNA segments are known to help in the control of genes, but most intergenic DNA has no

known function. Intergenic regions constitute about 75% of the human genome, 15% of bacterial genomes, and 30% of yeast genomes.

internet A system of linked computer networks used for the transmission of files and messages between hosts. Also see *intranet* and *local area network*.

interolog An evolutionarily conserved protein interaction identified between two pairs of orthologous proteins, such that A and A' and B and B' are each orthologs, with A interacting with B in one organism while A' interacts with B' in another organism.

intranet A computer network internal to a company or organization. Intranets are often not connected to the internet or are protected by a firewall. Also see *internet* and *local area network*.

intron From *intragenic region*. The part of a primary RNA transcript that is removed by splicing and, therefore, does not appear in the final RNA product.

IP address The unique, numeric address of a computer host on the internet.

isoelectric point (pI) The pH value at which the net charge of a protein or peptide is neutral, as determined by isoelectric focusing.

isotope coded affinity tagging (ICAT) A method used to directly compare the quantities of proteins from two different cellular states by mass spectrometry analysis of a shotgun digest of a proteome and appropriate isotopic labeling.

isobaric tags for relative and absolute quantification (iTRAQ) Multiplexed isobaric labeling method used in quantitative proteomics to determine the abundance of peptides and their cognate proteins across two or more biological samples.

IUPAC International Chemical Identifier (InChI) A text-based identifier for uniquely and unambiguously tagging or identifying chemical substances, designed to provide a standard, human-readable way to encode molecular information.

k-mers Collections of DNA "words" of length k observed in a sequence. Often used for rapid comparisons of large numbers of sequences.

label-free quantification A method used in quantitative proteomics to compare differences in the ion signal intensity or spectral counts of proteins and peptides between two or more experimental conditions.

library In sequencing, a collection of insert-containing clones. Sequencing libraries are created from a sequencing vector (see *plasmid*) and a set of inserts obtained by fragmentation of a larger piece of DNA.

linkage Genes or genetic markers that are physically close to one another on a chromosome and that tend to be inherited together.

linkage disequilibrium A state resulting when alleles at two defined loci are linked more frequently than would be expected, based on the known allele frequencies and recombination rate between the two loci. Linkage disequilibrium indicates that the two alleles being examined are physically close.

liquid chromatography A method for separating sample proteins or peptides in preparation for mass spectrometry; separations may be based on properties such as hydrophobicity (reverse phase), surface charge (ion exchange), or diffusion rate (size exclusion/gel filtration).

local area network (LAN) A network that connects computers in a small, defined area, such as the offices in a single wing or a group of buildings. Also see *internet* and *intranet*.

LOD score Short for *log odds*, a statistical estimate of the linkage between two loci on the same chromosome.

logo (or sequence logo) Graphical representation of a motif in DNA/RNA or protein sequences. Various methods are used but, commonly, the height at any position represents the information content at that position, and each letter's height is in proportion to its probability.

long terminal repeat (LTR) Identical sequences of DNA that repeat hundreds or thousands of times found at either end of retrotransposons or proviral DNA formed by reverse transcription of retroviral DNA. LTRs are used by viruses to insert their genetic

material into host genomes. LTRs resemble mobile gene-remodeling platforms that supply promoters and first exons for many eukaryotic genes.

low-complexity region Regions of biased composition, usually homopolymeric runs, short-period repeats, or the subtle over-representation of several residues in a sequence.

machine learning A branch of computer science that uses statistical techniques to give computers the ability to learn specific tasks, identify patterns, or make predictions using data without being explicitly programmed to do so. Machine learning allows scientists to produce reliable and repeatable decisions or results and uncover hidden insights by extracting trends within datasets.

marker genes Genes (including RNA genes) that can be used to distinguish taxonomic groups in a sample.

masking The technique by which low-complexity regions are removed from protein sequences, or LINE, SINE, Alu, and similar sequences are removed from nucleotide sequences prior to database searches.

mass accuracy Closeness of agreement between the result of a measurement and the true value (exact mass).

mass precision Closeness of agreement between independent mass measurement results.

mass range Minimum and maximum mass to charge ratio of ions that can be determined using a mass spectrometer. Also see *mass to charge ratio*.

mass resolution Ability of the mass spectrometer to differentiate between ions with close mass to charge ratios. Mass spectrometers with high resolution are capable of accurately detecting heavy isotope derivatives and the monoisotopic mass of a molecular ion. Also see *mass to charge ratio*.

mass spectrometry A collection of exquisitely sensitive and accurate analytical techniques that precisely measure molecular masses through a process of ionization and subsequent mass to charge measurement. Also see *mass to charge ratio*.

matrix-assisted laser-desorption ionization (MALDI) A common method for generating ions from analyte molecules for analysis by mass spectrometry.

messenger RNA (mRNA) A molecule in cells that carries codes from the DNA in the nucleus to the sites of protein synthesis in the cytoplasm (the ribosomes).

metabolite A small molecule that is an intermediate or a product of metabolism. Most metabolites have a molecular weight of <1500 Da. Primary metabolites are compounds directly involved in growth or cell division, while secondary metabolites are compounds that have an ecological, cell signaling, or protective function.

metabolome The collection of small-molecule metabolites found within a given biological sample (e.g. a cell, tissue, or organ) or within an organism under a particular set of conditions.

metagenome The set of genetic material isolated from a given habitat.

meta-omics A combination of approaches that can combine DNA sequencing, protein and metabolite characterization, and other techniques to characterize the microbiome in multiple and complementary ways.

microbiome The set of microorganisms associated with a given habitat.

microsatellite Regions of DNA containing short tandem repeats of a simple nucleotide sequence.

missense mutation A point mutation that changes a codon in such a way that a different amino acid is now encoded by that codon. If the amino acid substitution is non-conservative, the mutation can have a significant effect on the structure or function of the encoded protein.

mmu one millimass unit, or thousandth of a dalton.

molecular clock The hypothesis that nucleotide or amino acid substitutions occur at more or less fixed rate over evolutionary time, like the slow ticking of a clock. It has been proposed that, given a calibration date and a constant molecular clock, the amount of

sequence divergence can be used to calculate the time that has elapsed since two molecules diverged.

molecular complex A stable complex of molecules functioning as a biological unit.

monoisotopic mass The base mass of a protein or peptide containing none of the rarer, higher mass isotopes in any of its constituent atoms.

monophyletic The set of organisms descended from a common evolutionary ancestor or ancestral group.

motif Relatively conserved sequences within proteins or DNA that usually correspond to structural or functional regions. RNA motifs may contain information about the structure. Motifs are often represented mathematically as position weight matrices and are often represented graphically by logos.

multiple testing correction A statistical technique for addressing the problem that arises when repeating the same statistical test many times, such as during pathway enrichment analysis, where significant *p* values could appear by chance. Correction reduces the chance of obtaining false-positive results.

multivariate statistics A branch of statistics that focuses on developing or using statistical techniques for analyzing two or more variables of interest. Most “omics” data require multivariate statistics as the number of variables being considered is often in the hundreds or thousands. Multivariate means multiple variable.

mutation An irreversible modification to a chromosome. Mutations can involve single bases or entire regions of a chromosome. Mutations can either be neutral (have no effect), harmful, or beneficial. As such, mutations drive evolutionary change.

mass to charge ratio (*m/z* ratio) Quantity obtained by dividing the mass of an ion by its charge.

natural selection An evolutionary process in which genotypes that confer reproductive success are increased, while those that reduce reproductive success are decreased.

negative selection Natural selection against an allele or genotype, reducing its population frequency through time.

neutral mass The actual mass, in daltons, of a measured protein or peptide after deconvolution and subtraction of any associated ion mass. It refers to the neutral (non-charged) state of the analyzed molecule.

node A bifurcating branch point in a phylogenetic tree.

non-coding DNA A region of DNA that does not code for a protein.

nonsense mutation A point mutation that results in a premature stop codon in a gene, leading to premature truncation of the encoded protein.

normalization A statistical method that removes sources of variation in a dataset.

nuclear Overhauser effect (NOE) A nuclear magnetic resonance (NMR) phenomenon that involves the transfer of nuclear magnetism from one atom to another atom across short distances. NOEs occur through space rather than through bonds. Interatomic distances measured through NOEs can be used to determine the structure of proteins and other macromolecules. NOEs are measured through an NMR technique called nuclear Overhauser effect spectroscopy, or NOESY.

nucleotide The basic component of both DNA and RNA. Nucleotides consist of a base (adenine, cytosine, guanine, or cytosine), a sugar molecule, and a phosphoric acid molecule.

oligo Short for *oligonucleotide*. A short, single-stranded DNA or RNA. Most often used as probes for the detection of complementary DNA or RNA. Also see *complementary DNA*.

OnO (or O&O) Short for *ordered and oriented*. The particular order and direction (complemented or uncomplemented) of each contig from an assembly is known and specified.

ontology A structured terminology that describes all the concepts within a domain.

- open reading frame (ORF)** A DNA sequence that has the potential to encode a protein sequence. An ORF begins with a start codon (ATG) and ends with a stop codon (TAA, TAG, or TGA) in most species. See also *codon*.
- operational taxonomic units (OTUs)** Groupings of organisms that are constructed without a reference taxonomic database. In the context of marker-gene analysis, OTUs often comprise sets of marker-gene sequences that share a high degree of similarity. Also see *marker genes*.
- orthologs** Homologous sequences are said to be orthologous when they are direct descendants of a sequence in the common ancestor – that is, without having undergone a gene duplication event. Also see *homologs* and *paralogs*.
- outgroup** A distantly related sequence (or group of sequences) not belonging to the group or clade whose evolutionary relationships are being investigated.
- PAM matrix** PAM (point accepted mutation) and BLOSUM (blocks substitution matrix) are matrices that define scores for each of the 210 possible amino acid substitutions. The scores are based on empirical substitution frequencies observed in alignments of database sequences and in general reflect similar physicochemical properties. For example, a substitution of leucine for isoleucine (two amino acids of similar hydrophobicity and size) will score higher than a substitution of leucine for glutamate.
- paralogs** Homologous sequences that have arisen because of a gene duplication event. Also see *homologs* and *orthologs*.
- pathway** A set of interactions among biological entities; these interactions may or may not be linear or ordered in any way. Usually defined by a perturbation to a biological system whose output can be measured experimentally.
- pathway enrichment analysis** A statistical technique used to identify pathways that are significantly represented in a gene list of interest.
- pedigree** A tree representation of a family (cohort) showing the relationships between members and the pattern of inheritance of a given trait.
- peptide mass fingerprint** A protein identification method that works by enzymatically digesting a protein to produce a distinctive fingerprint of masses. The fingerprint is matched against putative fingerprints from protein or nucleotide databases to identify the unknown.
- phased sequence data** DNA sequence in which haplotypes are determined either by statistical estimation or by analyzing sequence data from the parents. By contrast, only the diploid genotype is known in unphased sequence data.
- peptide spectrum match (PSM)** Number of identified spectra matched to a given protein. The total spectral count can be higher than the number of peptides identified as peptides may be repeatedly identified by mass spectrometry.
- phenotype** The outwardly observable characteristics of an organism. Also see *genotype*.
- phylogenetic profile** A profile capturing the existence of orthologs of a gene across genomes. Genes with similar phylogenetic profiles are hypothesized to physically interact or at least be linked functionally.
- physical map** A genome map showing the exact location of genes and markers. The highest resolution physical map is the DNA sequence itself.
- plasmid** A circular, self-replicating piece of bacterial DNA. Numerous artificially designed plasmids contain priming sites, making them suitable for cloning and sequencing segments of DNA that range from 2000 to 10 000 bases in length.
- platform** Properly, the operating system running software on a computer, e.g. UNIX or Windows. More often used to refer to the type of computer, such as a Macintosh or PC.
- point mutation** Any mutation in which a single base pair is altered. Also see *missense mutation* and *nonsense mutation*.
- polymorphism** Common differences in DNA sequence among individuals that can be used as markers for linkage analysis. Typically, this refers to a locus in which the minor (less frequent) allele has a frequency greater than 1%. Also see *linkage*.

- polyphyletic** Sequences that may share some similarity but are not related by a common ancestor.
- positional cloning** Relies on the identification of a gene through pedigree analysis, genetic and physical mapping, and mutation analysis. Does not require extensive knowledge of the biochemistry of the disease in order to determine the gene responsible for the disease. The opposite of functional cloning.
- positive selection** Natural selection that favors an allele or genotype, increasing its frequency through time.
- post-translational modification (PTM)** Covalent modification of proteins after synthesis, usually involving a specific enzyme adding a chemical group to a specific amino acid (e.g. phosphorylation of serine, threonine, or tyrosine).
- Precision Medicine Initiative (PMI)** A U.S. national effort to enroll and track a volunteer participant cohort of 1 000 000 Americans, called “All of Us.”
- predictive analytics** Computer-based approaches for making prediction using data.
- primary structure** The linear sequence of a protein or RNA.
- primary transcript** The RNA molecule resulting from the transcription of the DNA sequence encoding a gene. In eukaryotes, it contains both the exons and introns prior to processing. Also see *exon* and *intron*.
- primer** An oligonucleotide used to initiate polymerase-mediated replication of a strand of DNA.
- principal component analysis** A statistical method that portrays a complex, multidimensional dataset (such as a genetic distance matrix) in a smaller number of dimensions.
- promoter** The region upstream of a gene where transcription is initiated.
- proteoforms** Different molecular forms in which the protein product of a single gene is found experimentally, encompassing various forms of genetic variation, alternative splicing of RNA transcripts, and post-translational modifications.
- pseudogenes** DNA that is similar to a normal, coding gene but that is not functional (may or may not be expressed). Pseudogenes are incapable of producing functional gene products. They are regarded as “genetic fossils” or defunct relatives of functional genes.
- p value** Also known as the probability value. It corresponds to the probability that the difference between two population or group means is simply due to chance. Hence a *p* value of 0.05 means that the two groups being evaluated have a 5% chance of being different because of random chance. A low *p* value (<0.05) for any kind of statistical comparison is usually a strong indication that the difference is significant.
- quantitative proteomics** A mass spectrometry technique for determining the amount of a given protein in a sample, generally used to compare differences in protein expression between two or more samples.
- quaternary structure** The arrangement or positioning of multiple polypeptide chains (with defined tertiary structures) in larger protein complexes.
- R factor** Residual disagreement. Used in X-ray crystallography as a measure of agreement between the experimentally measured diffraction amplitudes and those calculated using the protein coordinates. Perfect agreement corresponds to an *R* factor of 0.0. Total disagreement corresponds to an *R* factor of 0.59. Most good quality protein structures have *R* factors between 0.15 and 0.20.
- Ramachandran plot** A scatterplot showing the disposition of backbone φ (phi) and ψ (psi) torsion angles for each residue in a protein or set of proteins. Certain combinations of phi and psi angles are strongly preferred or are repeated over a series of residues, and these patterns can be easily detected in a Ramachandran plot.
- random forests** Also random decision forests. Machine learning methods for classification and regression that operate by constructing thousands of decision trees during training, then selecting and pruning the best tree(s) for performing the classification task in an optimal fashion. Random forest approaches are useful in multivariate statistics.

- repetitive DNA** DNA sequences of variable length that occur in multiple copies in the human and other eukaryotic genomes.
- resistome** The collection of all antibiotic resistance genes and their evolutionary precursors in both pathogenic and non-pathogenic bacteria.
- restriction fingerprint** The sizes of the DNA fragments resulting from an endonuclease digestion of the piece of DNA of interest.
- retention time (RT)** Measure of the time taken for a particular protein/peptide to pass through a chromatography column. It is calculated as the time from injection through to detection via mass spectrometry.
- retrotransposon** A transposon whose sequence shows homology with that of a retrovirus. Retrotransposons are a class of transposons that contain long terminal repeats and use an RNA intermediate for transposition. About 40% of the DNA in most mammalian genomes comprises retrotransposons. Also see *long terminal repeat*.
- ribosomal RNA (rRNA)** The RNA component of the ribosome, accounting for 60% of the ribosomal mass. Ribosomes are the engines for protein synthesis and are essential for all living organisms. Ribosomal RNAs are very abundant and constitute 80–90% of total cellular RNA.
- ribozyme** A catalytic RNA sequence.
- RNA interference (RNAi)** Also RNA-mediated interference. A cellular process in which double-stranded RNAs interfere with the expression of homologous genes through the degradation of complementary mRNA molecules. A technique commonly used to selectively suppress the expression of individual genes.
- root mean square deviation (RMSD)** An archaic term for standard deviation. RMSD is still used in the quantification of the atomic position differences between protein structures. Very similar structures have RMSD values between 0 and 1.5 Å; moderately similar structures have RMSD values between 1.5 and 3.0 Å.
- scaffold** See *supercontig*.
- secondary structure** In proteins, the local, regular backbone structures found in folded proteins (alpha helices and beta strands). In RNA, secondary structure is the set of canonical base pairs.
- selective sweep** The increase of allele frequency from low to high, owing to the effects of natural selection.
- sequence** In RNA, a sequence is an ordered arrangement of nucleotides. Sequences have directionality defined by the phosphodiester backbone and are customarily written from the 5' to 3' direction.
- sequence polymorphisms** Differences in DNA sequences that occur naturally among individuals. Also see *single nucleotide polymorphism*.
- sequenced tagged site (STS)** An operationally unique sequence that identifies the combination of primer pairs used in a polymerase chain reaction (PCR) assay, generating a reagent that maps to a single position within the genome. STSs are usually of the order of 200–500 bases in length.
- server** A computer that processes requests issued from remote locations by client machines.
- shotgun proteomics** An approach to proteome analysis where all components of a sample are first enzymatically digested into peptides, then separated by liquid chromatography, and finally analyzed by tandem mass spectrometry (MS/MS) to identify them. Also see *tandem mass spectrometry*.
- silencer** A DNA sequence capable of binding transcription regulation factors called *repressors*. Silencers prevent genes from being expressed as proteins. In other words, a silencer is a sequence-specific element that induces a negative effect on the transcription of an associated gene. Most silencers are found 20–2000 bp upstream of a gene.
- silent mutation** Point mutations that do not result in a change in the sequence of a protein.
- similarity** A quantitative measure of how related two sequences are to one another, usually assessed as the total number of identities and conservative substitutions in a pairwise

sequence alignment. Similarity does not automatically imply homology. Also see *alignment* and *conservative substitution*.

simplified molecular-input line-entry system (SMILES) A form of line notation for describing the structure of chemical species using short ASCII strings (i.e. alphanumeric characters that can be typed on a keyboard). Also see *ASCII*.

single amino acid variant (SAV) See *missense mutation*.

single nucleotide polymorphism (SNP) Alleles that are represented by single base changes in DNA sequence.

single nucleotide variant (SNV) A nucleotide position at which the base pair composition varies among individuals in a population.

single reaction monitoring (SRM) A tandem mass spectrometry method in which a single precursor ion of a specific mass is selected for the second round of fragmentation. Also see *tandem mass spectrometry*.

site An individual column of residues in an amino acid or nucleotide alignment. The residues at a site are presumed to be homologous.

spam Postings to newsgroups or mail broadcast to a large number of e-mail accounts which are usually wholly irrelevant or not of interest to the recipients. Analogous to postal junk mail.

stable isotope labeling of amino acids in a cell culture (SILAC) An *in vivo* metabolic isotope labeling technique that incorporates heavy and light isotopic forms of amino acids into newly synthesized proteins.

subcellular localization The cellular compartment to which a protein is transported and where it performs its intended function.

supercontig A stretch of DNA sequence composed of one or more contigs with known order and orientation.

support vector machine (SVM) A supervised machine learning method that can be used for both classification and regression tasks. SVM methods find an optimal hyperplane in hyperdimensional space to separate or classify objects that are described by many variables. SVMs are useful in multivariate statistics.

synapomorphy A characteristic present in an ancestral species and shared exclusively by its evolutionary descendants.

synteny In comparative mapping, the observation that the order of loci in a chromosomal region of one organism is conserved in a chromosomal region of a second organism. From the Greek for “on the same ribbon” (*σύν*, with + *ταινία*, ribbon).

tandem mass spectrometry (MS/MS) A process whereby a first stage of mass spectrometry is used to select certain components of a sample, which are then broken down for further analysis by a second stage of mass spectrometry. In some instruments this can be applied repeatedly to yield MSⁿ separations. Also see *mass spectrometry*.

taxon Any named group of organisms. They do not need to form a clade.

taxonomic profiling The use of bioinformatic techniques to map DNA sequences to reference databases with associated taxonomic information in order to get qualitative and quantitative estimates of the biodiversity in a sample.

telomere A region of repetitive nucleotide sequences at each end of most eukaryotic chromosomes that protects the ends of the chromosomes from deterioration or from fusion with neighboring chromosomes. Telomeres compensate for incomplete semi-conservative DNA replication at chromosomal ends.

tertiary structure The arrangement or positioning of secondary structure elements into compact, non-overlapping globules or domains. Tertiary structures are the three-dimensional structures of proteins. In RNA, tertiary structure is the three-dimensional arrangement of atoms.

terminator A section of DNA that marks the end of a gene or operon during transcription. This sequence mediates transcriptional termination, providing signals in the newly

synthesized transcript RNA that trigger processes that release the transcript RNA from the transcriptional complex.

threading A method for predicting the most likely fold or topology of a protein by assessing the likelihood that its sequence “fits” into a known three-dimensional fold or a known arrangement of secondary structure.

Thomson unit (Th) Unit of mass to charge ratio relating to the field of mass spectrometry. Also see *mass to charge ratio*.

tiling path format Format indicating the identities and order of sequences to be included in a targeted region assembly.

time-of-flight mass spectrometer (TOF-MS) A mass spectrometer that measures mass to charge ratios by the time required to traverse a set distance. Also see *mass to charge ratio*.

top-down proteomics (TDP) A proteomic method that begins with mass spectrometry (MS) of intact proteins, followed by subsequent analysis of their constituents via MS degradation methods or peptide mass fingerprinting. Also see *mass spectrometry*.

topology The map or plan of a physical system or set of connected objects. The topology of proteins is generally described by their backbone tertiary (three-dimensional) structure. In phylogenetics, the branching pattern of a tree. In transmembrane proteins, the sequence of non-membrane segments on either side of the membrane.

transcription start site (TSS) The location where transcription starts at the 5' end of a gene sequence. The TSS is always upstream of the translation initiation site (TIS) and often includes an RNA polymerase or transcription factor binding site. Knowledge of the exact position of a 5' TSS of an RNA molecule is crucial for the identification of the regulatory regions that immediately flank it. Also see *translation initiation site*.

transfer RNA (tRNA) An adaptor molecule composed of RNA (typically 76–90 nucleotides in length) that serves as the physical link between the mRNA and the amino acid sequence of proteins. tRNAs carry amino acids to the ribosome, acting as molecular “translators” that match the codon in an mRNA with the amino acid for which it codes.

translation initiation site (TIS) The site on a given mRNA (and its corresponding DNA) where the ribosome binds to start translation, the conversion of mRNA codons into proteins. Most TISs begin with an ATG start codon but there are many situations where this is not the case.

transmembrane protein A protein that contains at least one segment which crosses through a biological membrane.

transposase An enzyme that binds to the end of a transposon and catalyzes the movement of the transposon to another part of the genome via a cut-and-paste mechanism.

untranslated region (UTR) A region of an mRNA molecule that is not translated to protein. There can be two types of UTRs: 5' UTRs and 3' UTRs. A 5' UTR is the untranslated portion of an mRNA stretching from the 5' end to the position of the first codon used in translation. The 3' UTR is the untranslated portion of an mRNA stretching from the 3' end of the mRNA to the position of the last codon used in translation.

user The person using client–server or other types of software.

Variant Call Format (VCF) A standard format for storing genetic data in plain text.

van der Waals volume The spherical space around an atom that cannot be occupied by another atom. In molecules such as proteins, the van der Waals volumes of multiple atoms form the van der Waals surface.

vertical gene transfer The movement of genetic material from parents to offspring. Also see *horizontal gene transfer*.

Viterbi algorithm A dynamic programming algorithm that allows one to compute the most probable or optimal path. It is similar to the Needleman–Wunsch algorithm used in pairwise sequence alignments. More formally, the Viterbi algorithm finds the most likely sequence of hidden states (called the Viterbi path) that results in a sequence of observed events.

wiki A web site that can be used for collaborative or informational purposes, in which any individual can freely add, edit, or delete information. The term originates from the Hawaiian word for quick (*wikiwiki*).

word matching Computer-based search for small segments (“words”) of identical DNA sequence.

yeast artificial chromosome (YAC) A vector used to clone segments of DNA up to 1 million bases in length.

Z score, Z value This measures the distance of a value from the mean of a normal or Gaussian distribution in standard deviation units. A *Z* score of 1 means the value is one standard deviation away from the mean. A *Z* score of 4 indicates the value is four standard deviations away from the mean (indicating the value has <99.9% chance of occurring randomly). A high *Z* score typically means a greater level of statistical significance.

Index

- 3D-JIGSAW 382
 1000 Genomes Project 485, 490, 495, 498
- a**
- acceptance 49
 AccessFold 175–6, 178
 accessible surface area (ASA) 189, 372
 ACCpro5 192
 ACD/ChemSketch 443
 ACeDB database 418
 Achilles visual data quality viewer 544
 activity flow (AF) diagrams 419
 adjacency matrix 426
 admixture 488, 490
 AED score 144
 affine gap penalty 51
 AffyBatch 284
 affyPLM package 284–5
 agglomeration methods 293
 Alifold 162, 178
 ALIGN 260
 alpha-diversity approaches 517
 AMDIS 456, 458
 amplicon sequence variant (ASV) approaches 514
 analysis of variance (ANOVA) 298, 459, 521, 578–9
 ANAREA algorithm 387
 Andromeda/MaxQuant 347
 annotated spectral library (ASL) 343
 annotation edit distance (AED) 143
 ANNOVAR 540
 ANOVA 298, 459, 521, 578–9
 Anscombe's quartet 564
 ARACNE 430
 ArrayExpress databases 280
 Artemis 145, 147
 artificial neural networks 121, 470
 ASAP 309
 ASEdb 211
 ASL 343
 association studies 538
 Atlas data browser 544
 AUGUSTUS 132–6, 144
 Automated Mass Spectral Deconvolution and Identification System (AMDIS) 456, 458
- b**
- Baas-Becking hypothesis 511
 background selection 493
 balancing selection 493–4
 BALiBASE 229–30, 234, 235, 237
 ball-and-stick representations 378
 base-pairing probability 170
 Basic Local Alignment Search Tool *see* BLAST
 BatchQC 291
 BATMAN 456
 Bayesian inference 264
 Bayesian statistics 456
 Bayesil web server 444, 455, 456
 BCFtools 486, 490, 491
 BeadLevelList 284
 BEAST 265
 Benchmarking Universal Single-Copy Orthologs (BUSCO) 135, 144
 Benjamini–Hochberg correction 420, 577
 beta-diversity 518–9
 BETAWARE 197, 198
 BID 211
 BiFold 175
 Binary Alignment/Map (BAM) 81, 82, 486
 binary classifications 128–30
 binary PLINK file (bfile) 491, 492
 binomial models, negative 298–9
 BIOML 347
 BioCarta 302, 450
 Bioconductor packages 284
 BioCyc database 405, 417, 446
 BioGRID 407, 411
 BioMagResBank (BMRB) 373, 446, 448
 BioMart 97, 108–10

- BiomeNet 528
 - BioPAX 416, 417, 450
 - Biopolymer Markup Language (BIOML) 347
 - BioVUK 545, 547
 - BLAST 21, 52–61, 80, 86, 140, 192, 206, 265, 270, 391, 577
 - algorithm 52–4, 92, 108, 526
 - search 53–8
 - suggested cut-offs 61
 - understanding the output 58–61
 - BLAST 2 Sequences 61–2
 - BLAT 66–70, 80, 81, 96, 135, 144
 - algorithm 92
 - search 91–93, 112
 - submitting a query 70
 - BLOCKS database 49
 - BLOSUM (Block Substitution Matrix) 49–50, 262
 - BMRB 373, 446, 448
 - BOCTOPUS2 198
 - Bonferroni correction 299, 577
 - bootstrapping 259, 294
 - bottom-up proteomics 329–30
 - Bowtie 2 283, 523, 525
 - BRAKER suite of programs 144
 - BRAunschweig ENzyme DAtabase (BRENDA) 354
 - Bray–Curtis dissimilarity (BCI) 519
 - BRENDA 354
 - browser extensible data (BED) 81, 82
 - Burrows–Wheeler aligner (BWA) 283, 525
 - BUSCO 135, 144
- C**
- CAFA 202, 207
 - CAGI 547, 548
 - Cancer Genome Anatomy Project (CGAP) 67
 - Cancer Genome Atlas, The (TCGA) 286
 - capillary electrophoresis (CE) 344
 - CASP 177, 195, 201, 206, 207, 331, 382, 385, 547
 - categorical random variable 557
 - CATH databases 203, 390–91
 - CAZy 527
 - CBLAST 377
 - CDD 58, 203, 377
 - cDNA sequencing 280
 - CEL-Seq 308
 - central limit theorem 567, 574, 575
 - centroid 325
 - CFM-ID database 448
 - CGView 142
 - charge state reduction 328
 - Chemical Entities of Biological Interest database (ChEBI) 444, 446, 448
 - Chemical Markup Language (CML) 441
 - Chemistry Development Kit (CDK) 440
 - ChemSpider 446–8
 - Chime 379
 - chimeric sequences 509
 - chi-squared test 579
 - Chorus 352
 - ChromaTOF 456, 458
 - chromatography 400
 - Chromopainter 490
 - CING 374
 - clade annotation 136
 - Cline shift score 230
 - Clinical Pharmacogenomics Implementation Consortium (CPIC) 538
 - Clinical Proteomic Tumor Analysis Consortium (CPTAC) 545
 - ClinicalTrials.gov 36
 - ClinVar database 538, 539
 - Clustal 232–40, 242–3, 245, 260, 261
 - ClusterMaker2 Cytoscape app 429
 - clusters of orthologous groups (COG) 203
 - CML 441
 - Cn3D 32, 377, 379
 - coalescence 487, 488
 - coarse-graining 177
 - coding sequences (CDSs) 2
 - coding statistics 124
 - coefficient of determination 561
 - co-evolutionary coupling 385, 386
 - COGIC 206
 - collisional fragmentation 319
 - collisional-activated dissociation (CAD) 319
 - collision-induced dissociation (CID) 319, 342
 - COMBAT 290
 - comparative sequence analysis 162, 163
 - Competitive Fragmentation Modeling and Identification (CFM-ID) database 448
 - composite of multiple signals (CMS) test 495
 - compositional similarity 510
 - Comprehensive Antibiotic Resistance Database 527
 - concordance index 305
 - conditional rarity 511
 - confidence interval (CI) 560
 - CONSENSE program 266
 - Conserved Domain Database (CDD) 58, 203, 377
 - CONTRAST 136
 - ContTest 231
 - CONVERTF 491

- correlation 561
 - correlation-based methods 429
 - COSMIC database 538
 - COSY 365
 - coulometric (electrochemical) array systems 439
 - covariance 561
 - CPHModels server 382
 - CPK (Corey, Pauling, and Koltun) model 378
 - Critical Assessment of (Protein) Structure Prediction *see* CASP
 - Critical Assessment of Function Annotation (CAFA) 202, 207
 - Critical Assessment of Genome Interpretation (CAGI) 547, 548
 - Crux software suite 346
 - Cufflinks 133
 - curated databases 11
 - Cyc databases 450, 471
 - Cytoscape 429–31, 443, 473
 - AutoAnnotate app. 425
 - CyNI toolkit app 430
 - network visualization 422
- d**
- DADA2 514–15
 - Dali 376, 391
 - Database for Annotation, Visualization and Integrated Discovery (DAVID) 301
 - Database of Genomic Structural Variation (dbVAR) 20
 - data-dependent acquisition (DDA) procedures 333
 - data-independent acquisition (DIA) 333
 - DAVID 301
 - dbNSFP annotation database 541
 - dbPTM 355
 - dbSNP 20, 28, 89, 113
 - DBS-PSSM 211
 - dbVAR 20
 - Database of Single Nucleotide Polymorphisms (dbSNP) 20, 28, 89, 113
 - de novo* peptide sequencing 341–2, 344
 - deconvolution 328
 - DeepLoc 209–210
 - DeepView 380–2, 389
 - degrees of freedom 573–4
 - descriptive statistics 558–61
 - DESeq 289, 298, 299, 528
 - Dfam 140
 - DIALIGN 237–8
 - DIAMOND 525
 - Dictionary of Secondary Structure for Proteins (DSSP) 189, 190, 192, 387
 - differential analysis 429
 - differential expression analysis for sequence count data 289, 298, 299, 529
 - differential expression testing 296–300, 304–6
 - diffusion approximations for demographic inference (*dadi*) 488
 - dihedral angles 369
 - dimensionality reduction 459
 - dimethyl labeling 332
 - Dirichlet distribution 120
 - discovery-based global profiling approaches 333
 - discriminant analysis 579
 - DISOPRED 200, 201
 - disorder prediction performance 201
 - disordered regions 199–201
 - DisProt 200
 - dN/dS ratio 257–8
 - DNA Databank of Japan (DDBJ) 2–4, 16, 19
 - DNA methylation 546
 - DNA sequencing 505, 506
 - DomCut 205
 - Dom-Pred 205, 206
 - DOMpro 205
 - DomSSEA 205
 - dotplot 61
 - DOUBLESCAN 136
 - DP-Bind 211
 - DPS 205
 - Drop-Seq 308
 - Drug Repurposing Hub (Broad Institute) 545
 - DrugBank 538
 - DSSP 189, 190, 192, 387–8
 - DuplexFold 175
 - DUST 140
 - Dynalign 162, 163, 170, 176, 178
 - dynamic mass range capability 317
 - dynamic programming 126, 159–60, 586–90
 - DyNet Cytoscape app 429
- e**
- E. coli* Metabolome Database (ECMDB) 450, 451
 - EasyGene 1.2 131
 - EasyGene 118, 123
 - EasyModeller 382
 - eBURST 268
 - ECMDB 450, 451
 - EcoCyc 401, 405, 407, 418
 - ecological relevance 511
 - edgeR 298, 299, 528

- Edinburgh Human Metabolic Network database 473
- Edman protein sequencing 316
- effect size 575, 577
- EGASP 128
- EggNOG database 206
- EHH test 494
- EIGENSOFT 490, 491
- EIGENSTRAT tool 485, 491
- electron capture dissociation (ECD) 319
- electron microscopy 364–5
- electron transfer dissociation (ETD) 319
- electron-based fragmentation 319
- electrospray ionization (ESI) 316
- electronic health records 537–8, 542–3
- ethical, legal, and social implications of translational medicine (ELSI) 549–50
- ELM 211
- European Nucleotide Archive (ENA) 2
- ENCODE 84, 88, 113, 132
- ENCODE Genome Annotation Assessment Project (EGASP) 128
- Encyclopedia of DNA Elements *see* ENCODE
- ENDEAVOUR 541
- Enrichment Map visualization software 422
- Ensembl 112, 144, 341, 354
- ENSEMBL Genome Browser 79–82, 85, 92, 96–108, 112, 145
- entity relationship (ER) diagrams 419
- Entrez 20–32
- Environmental microbiomes 505, 506
- ESI 316, 317
- EST (Expressed Sequence Tag) 4
- eThread 385
- Euclidean distance 292, 305
- Eukaryotic Annotation Pipeline 143
- European Bioinformatics Institute (EBI) 2, 80, 450
- European Macromolecular Structure Database (MSD-EBI) 373
- European Molecular Biology Laboratory (EMBL) 1, 2, 16, 19, 145, 373
- European Nucleotide Archive (ENA) 2
- EVA 195
- EVcomplex 211
- EVfold 211
- EvidenceModeler (EVM) 143
- Exome Aggregation Consortium (ExAc) 486
- Exonerate 134, 135
- expected heterozygosity 483
- Expressed Sequence Tags (ESTs) 4
- expression analysis 279–310
- classification methods 306–8
 - classifier 302–6
 - data pre-processing 283–4
 - data/metadata collection and management 282–3
 - differential expression analysis 296–300
 - DNA microarrays 280
 - experimental design 281–2
 - exploratory data analysis 291–6
 - functional enrichment analysis 300–302
 - normalization and batch effects 287–91
 - quality control 284–6
 - single-cell sequencing 308–9
 - validation of predictive models 306–8
- expression matrix 287
- ExpressionSet 284
- extended haplotype homozygosity (EHH) test 494
- ExtensibleMarkup Language *see* XML
- extrinsic (evidence-based) gene finders 132
- f**
- F1 score 129
- Falco 309
- false discovery rate (FDR) 298, 336, 521, 576
- family-wise error rate (FWER) 299, 577
- Fast Healthcare Interoperability Resources (FHIR) 544, 548
- FASTA 3, 70–75, 145, 170, 233
- FastQC 285, 439
- FASTQ-formatted file 509
- fastSTRUCTURE 490
- FastTree-2 240
- FATCAT 376, 389, 391–2
- feature table
- in ENA Format 584–5
 - in GenBank/DBJ Format 585–6
- FFPred 206–7
- Fiehn Metabolome Database (BinBase) 458
- File Chameleon 97
- FILM3 211
- fineSTRUCTURE 490
- Fisher's exact test 301, 420, 472
- FIT 378
- Fitch–Margoliash (FM) 263
- five-kingdom classification system (Whittaker) 252
- fixation index (F_{ST}), 483, 484
- flatfile 3
- flatfile header
- in DDBJ/GenBank Format 583–4
 - in ENA Format 583

- fluorescence resonance energy transfer (FRET) 556
 fluorescent in situ hybridization (FISH) 529
 FlyBase 341
 FOAM 525
 FoldAlign 162, 163, 176, 178
 Formula Predictor (Shimadzu) 459
 Fourier transform infrared spectrometers (FTIR) 439
 Fourier transform ion cyclotron resonance (FT-ICR) spectrometer 317, 459
 fragments per kilobase million (FPKM) 287–8
 Frappe 490
 free energy minimization 170
 Frequency and Probability Distributions 566–8
 FRODO 378
F-test 298
 functional enrichment analysis 300–302
 functional interaction databases 410–14
 Functional Ontology Assignments for Metagenomes (FOAM) 525
 FunctionSpace 206
 FunFams 207
- g**
- g:Profiler 421, 423
 Galaxy-M 460, 468
 gamma distribution 258
 gap penalties 51–2
 gaps 51–2
 gas chromatography (GC) systems 439
 Gaussian filtering 325
 Gaussian fit 325
 Gaussian subtype classification models 306
 GC-MS-Based Compound Identification 456–8
 gel-based separation techniques 400
 gel-eluted liquid fraction entrapment (GELFrEE) 344
 gelML 352
 GenBank 1–4, 16, 19, 20, 30, 31, 38, 50, 79, 131, 145, 254, 373, 444, 450
 GENCODE 80, 84–8, 113, 141
 gene annotation and evidence generation
 finding/removing pseudogenes in eukaryotes 141
 prophage finding in prokaryotes 137–8
 repetitive sequence finding/masking in eukaryotes 138–41
 tRNA and rRNA gene finding 136–7
 using comparative gene prediction 135–41
 using protein sequence databases 134–5
 using RNA-seq data 133–4
 gene co-expression analysis 402
 gene duplication 255
 Gene Expression Omnibus (GEO) 280
 gene-finding programs 118
 gene flow 482
 gene fusion method 402
 gene neighborhood 402, 412
 Gene Ontology (GO) 14, 283, 301, 425, 527, 543
 prediction of 201–2
 Gene Ontology annotation (GOA) 9
 gene prediction, *ab initio* 118
 evaluation 127–30
 in eukaryotic genomes 123–4, 131–3
 in prokaryotic genomes 118–23, 131
 Gene Set Enrichment Analysis (GSEA) 283, 302, 421, 424, 473
 GeneID 132, 134, 136
 Geneious 265
 GeneMANIA 414–15, 414, 431, 542
 GeneMark 118, 123, 131, 132, 144, 525
 Generic Model Organism Database (GMOD) project 111
 genetic drift 482, 493
 GeneWise 134, 135
 GeneZilla 132
 GenGIS 270
 GENIE 126
 Genome Aggregation Database (gnomAD) 486
 genome annotation 117–46
 evidence generation for 133–5
 genome annotation pipelines 141–5
 eukaryotic 142–4
 prokaryotic 142
 visualization and quality control 145
 Genome Reference Consortium (GRC) 80–2, 84
 GenomeScan 132, 134
 GenomeThreader 135
 GenomeView 145
 genome-wide association studies (GWAS) 485, 545, 577
 genomic data, file types 82–4
 Genotype-Tissue Expression (GTEx) project 91, 92
 GENSCAN 118, 126, 132, 136, 141
 Gibbs free energy 157–8
 GiniClust/GiniClust2 309
 GLIMMER 118, 123, 131
 global alignment 255
 global NMR data center (BMRB) 373

- Global Proteome Machine (GPM) 343, 353–4, 358
- global sequence alignment 46
- GlobalAncova 472
- GLOBETROTTER 490
- GMAP 144
- Gnomon gene-finding program 144
- Goldberg–Hogness box 124
- Golm Metabolome Database 446, 448
- GOR 187
- graph theory 425–6
- graphical interfaces 9
- Greengenes 515
- GSEAlm 302
- GSNAP 133
- GSS (Genome Survey Sequences) 4
- guide tree 228, 260, 261
- guilt-by-association (“guilt-by-correlation”) 356, 427, 541
- GutenTag 341
- h**
- HAMAP 203
- hard masking programs 140
- hard sweep 493
- Hardy–Weinberg equilibrium (HWE) 483, 489
- Health Insurance Portability and Accountability Act of 1996 (HIPAA) 549–50
- heat map 291–2, 293
- Henderson, Richard 367
- heterozygosity 483
- heterozygote advantage 493
- heuristics 228
- HHblits 188, 193
- HHpred 382, 384
- HHsearch 202
- hidden Markov models *see* HMMs
- hierarchical clustering 291–4
- higher energy collisional activation dissociation (HCD) 319
- high-performance liquid chromatography (HPLC) 322
- histogram 562
- histone marks 89, 104–5
- HMDB 444, 458, 459, 470
- HMMER 135, 202, 234, 240
- HMMs 64, 121, 122, 125, 187–9, 197, 233, 234, 525, 527
- HMQC 365
- hole filling 401
- homology 45, 46
- homology modeling 382–3
- Homology-derived Secondary Structure of Proteins (HSSP) database 190
- HomPPI 211
- HomPRIP 211
- Homstrad structure alignment database 230
- horizontal (or lateral) gene transfer 255
- hot spots 210
- HTML5 Molecular Editor 443
- Human and Vertebrate Analysis and Annotation (HAVANA) group 85
- Human Gene Mutation Database (HGMD) 538
- Human Genome Project 19, 66
- Human Metabolome Database (HMDB) 444, 446, 450–51, 448
- human microbiome 505–6
- Human Microbiome Project 506
- Human Phenotype Ontology (HPO) 543, 544
- Human Proteome Organization (HUPO) 348
- hybrid search 343
- hydrophobic liquid interaction chromatography (HLIC) 344
- hydrophobicity 196
- hypergeometric test *see* Fisher’s exact test
- hypothesis-driven directed approaches 333
- i**
- iCn3D 32, 377, 379
- ID line 4
- IDEAL 200
- immobilized metal ion affinity chromatography (IMAC) 324
- InChI strings (International Chemical Identifier) 440
- Inferred Biomolecular Interactions Server (IBIS) 211, 377
- infrared multi-photon dissociation (IRMPD) 319
- Ingenuity Pathway Analysis 450
- InsPecT 341
- IntAct 407, 412
- integrated haplotype score (iHS) 494, 499
- Integrated Metabolomic and Expression Analysis (INMEX) 473
- integrative analysis 430
- International Classification of Diseases (ICD) 543
- International Nucleotide Sequence Database Collaboration (INSDC) 2, 12
- International Society for Biocuration 11
- International Union of Pure and Applied Chemistry (IUPAC) 439

InterPro 8, 202, 390
 InterProScan 202
 interquartile range (IQR) 562
 intrinsically disordered proteins (IDPs) 199
 intrinsically disordered regions (IDRs) 199
 intrinsically unstructured proteins 199
 Invitae 540
 ion trap mass analyzer 322
 IPfam 211
 iProX 352
 iRefIndex 414
 isoelectric focusing (IEF) 344
 isotope-coded affinity tagging (ICAT) 331
 isotopic dilution analysis 459
 I-TASSER 385
 IUPAC nomenclature rules 439

j

Jaccard index 519
 jack-knife technique 294
 Jalview 231, 232, 243
 JavaScript viewers 374
 JBrowse 81, 110–12, 145
 JC69 model 262
 JCAMP-DX 441, 443
 JChemPaint 443
 J-couplings 365
 JDXview 443
 jfpred 207
 JIGSAW 143
 JME 443
 Jmol 389, 392, 443, 444
 jPOST 352
 JSME Molecule Editor 443
 JSmol 374, 376, 443
 JSpectraViewer 444, 445
 JSpecView 443
 JuiceScreener 456

k

K80 model 262
 Kalign 237, 238
 Kallisto 283, 309
 Karlin–Altschul equation 54
 KEGG 302, 354, 406, 408, 444, 446, 450, 471, 473, 525, 527
 KGML (KEGG Markup Language) 450
 Kingdon trap 318
 Kiosk Viewer 374, 379
k-mer decomposition 516, 526
 KNApSACk 446, 448
k-nearest neighbors model 306
 KnowItAll Academic 443

knowledge 555–7
 Kolmogorov–Smirnov test 302
 Kozak consensus sequence 124
 Kraken 526
 Kruskal–Wallis test 578
 Kyoto Encyclopedia of Genes and Genomes *see*
 KEGG

l

ladder of nature (*scala naturae*) 252
 Lagrangian multipliers 389
 last universal common ancestor (LUCA) 251
 leaderless transcription 121
 LEfSe 521, 522
 level of significance 573, 574
 LIGAND 406
 Ligand Explorer 374, 379
 Limma 298
 linear algebra 426
 linear discriminant analysis 306
 linear splines 325
 LIPID MAPS 444, 446–8
 liquid chromatography (LC) 439
 liquid chromatography-mass spectrometry
 (LC-MS) 439, 458–9
 Livebench 195
 local sequence alignment 46, 255
 LocARNA 163, 176, 178
 LocTree3 209
 Loess derivative filters 325
 log likelihood scoring matrix 121
 log odds ratio (lod score) 48
 logistic regression 211, 306
 log-normal distribution 567
 LOMETS (Local Meta-Threading Server) 385
 long branch attraction problem 259
 loop initiation energies 159
 LOOPP 385
 LTR_FINDER 140
 LTRharvest 140

m

machine learning approaches 521
 Macromolecular Transmission Format
 (MMTF) 375
 MAFFT 229, 234, 237–40
 MAKER2 143, 144
 Manhattan plot 495
 mapping-first approach 133
 marker gene analysis 511–521
 associations with metadata 521
 calculating and comparing diversity 516–20
 general considerations 509–11

- marker gene analysis (*contd.*)
 - grouping of similar sequences 513–15
 - quality control 513
 - ribosomal RNA genes 512–13
 - taxonomic assignment 515–16
- Markov chain Monte Carlo (MCMC) algorithm 264, 265, 489
- Markov cluster (MCL) algorithm 426
- Markov models 121–2
 - codon position-dependent fifth-order 125–6
- MarvinSketch 443
- MarvinView 444
- Mascot 336–8, 341–2, 348–9
- mass spectrometry (MS) 315–56, 399, 439 *see also* tandem mass spectrometry
 - ion detectors 321
 - ionization 317
 - mass analyzers 317–20
 - proteomics 132–3, 325–8
- Mass Spectrometry Interactive Virtual Environment (MassIVE) 352
- mass spectrum 321
- MassBank of North America (MoNA) 446, 448, 449
- MassHunter (Agilent) 458, 459
- MassIVE 352
- MassLynx (Waters) 458
- match factor 457
- MATLAB 460
- matrix diagonalization techniques 389
- matrix-assisted laser desorption ionization (MALDI) techniques 316, 317, 328
- Matthews correlation coefficient (MCC; CC) 128, 129
- MAVEN 460
- Max Planck Bioinformatics Toolkit 232
- maximal dependence decomposition 126
- maximum expected accuracy method 169, 170
- Maximum Likelihood (ML) methods 264
- Maximum Parsimony (MP) 264
- MaxQuant (Andromeda) 341, 345, 347
- mBed algorithm 232
- MC-Fold 176
- MC-SYM software 176
- mean filtering 325
- MED 2.0 131
- median 560
- median filtering 325
- medical databases 33–8
- medical subject headings (MeSH) 543
- MEDLINE layout 26
- MEGA 265
- MegaBLAST 62–4, 526
- MEGAN 527
- MeltDB 460
- MEMSAT-SVM 197, 198
- MetaboAnalyst 460–5, 468, 470–3
- metabolic flux balance analysis 473
- metabolic reconstructions 473
- metabolic simulations 473
- MetaboLights 444, 446, 450
- metabolite Identification 451–9
 - GC-MS-based compound identification 456–8
 - LC-MS-based compound identification 458–9
 - NMR-based compound identification 454–6
 - targeted versus untargeted metabolomics 453
- metabolite interpretation 470–73
- metabolite levels 437
- metabolite set enrichment analysis (MSEA) 472
- metabolome 437
- metabolomics 272, 437–73, 529
 - chemical compound databases 444–8
 - chemical representation and exchange formats 440–41
 - data formats 439–44
 - metabolic pathway databases 449–50
 - meta-metabolomics 508, 529
 - molecular editors 442–3
 - organism-specific databases 450–51
 - spectral databases 448
 - spectral representation and exchange formats 441–2
 - spectral viewers 443–4
 - targeted versus untargeted 453
- Metabolomics Standards Initiative (MSI) 453
- Metabolomics Workbench 446, 450
- MetaCyc 405, 446
- metagene 294–6
- MetaGeneMark 525
- MEtaGenome ANalyzer (MEGAN) 523
- metagenomic and meta-metabolomic, combined analysis 529
- metagenomic data analysis 505, 521–8
 - assembly 524
 - functional predictions 527–8
 - gene annotation and homology searching 524–5
 - metagenomic workflow 508

- predicting functional information from
 - marker-gene data 522–3
- protocol 523
- quality control and merging of paired-end reads 523–4
- statistical associations 528
- taxonomic assignment and profiling 525–7
- metagenomeSeq 517
- metagenomic sequence analysis 505
- metagenomics 272, 528–9
- MetaMap 544
- MetaMapp 473
- MetaMapR 473
- metAMOS 523
- Meta-P server 460
- MetaPhlAn 526
- metaprdos2 200, 201
- metaproteomics 507, 529, 529
- metaSPAdes 524
- Metastudent 206
- metatranscriptomics 507, 529, 530
- methyl-seq 285
- METLIN 444, 446, 448, 459
- MetPA 472
- MetScape 473
- MFO 207
- Mfold 156, 163, 164, 177
- mGENE 133, 134
- MIAPE-MS 348
- MIASSPE 348
- microbiome analysis 505–530
- Microbiome Helper 523
- MicroReact 270
- minimum evolution (ME) 263
- minimum redundancy maximum relevance (mRMR) 304, 305
- MinPath 528
- MITE-Hunter 140
- mMass 443–4
- MobiDB 203
- ModBase 382
- mode 559–60
- MODELLER 382
- ModerNA program 177
- ModWeb server 382
- Molecular ACCess System (MACCS 166) 440
- molecular clock hypothesis 253–4
- molecular cross-linking 401
- Molecular Design Limited (MDL) 440
 - chemical fingerprint 440
 - Information Systems 378
- molecular dynamics (MD) simulations 177
- molecular editors 442–3
- molecular evolution see phylogenetic analysis
- molecular interaction databases 407–10
- Molecular Modeling Database (MMDB) 21, 22, 33, 377
- Molfile (MOL) 441
- MolProbity 373, 387, 388
- MoNA 459
- Monarch Initiative 542
- monoisotopic mass 328
- monotonic relationship 561
- Monte Carlo (random search) methods 21, 385
- mothur 513
- Mouse Genome Database (MGD) 11, 39, 41, 43
- Mouse Genome Informatics (MGI) resource 39, 40
- MOWSE probability algorithm 336
- mpileup (Samtools) 486
- MrBAYES 265
- mRMR package 305
- MS PepSearch (NIST) 343
- MSA 202, 232
- MS-Align+ 344
- MS-DIAL 458–60
- MSEA 473
- msf 233
- MS-GF+ 341, 345
- mtDNAProfiler 269
- Mugsy-Annotator 136
- multidimensional distributions 568
- multidimensional scaling 484, 519, 520
- Multilign 170, 175, 176, 176
- MultiLoc2 209–210
 - HighRes 209
 - LowRes 209
- multi-locus sequence typing (MLST) 268
- multimodal distribution 567–8
- multi-omic datasets 529
- multiple displacement amplification (MDA) 529
- multiple enrichment analysis 424
- multiple reaction monitoring (MRM) 354, 459
- multiple sequence alignment (MSA) 45, 64, 120, 187–8, 227–46, 255, 385
 - building 231
 - profile 135
 - quality, measuring 228–31
 - viewing 242–6
- multiple sequentially Markovian coalescent (MSMC) method 486–7, 489, 499
- multiple threading servers 385
- MultiQC 285

- multivariate statistics 459–60
 - MUSCLE 229, 230, 236, 238–40, 260–1
 - MUSTANG 229
 - MUSTER 385
 - mutation 481, 493
 - MutPred 541
 - MVAPACK 460
 - Myriad Genetics 540
 - MySQL database 353
 - MyVariant.info 538
 - MzCloud 448
 - mzData 352
 - mzIdentML 352
 - MZmine 458, 459
 - mzML 348, 441, 443
 - mzQuantML 352
 - mzTab 352
 - mzXML standards 352
- n**
- N50 524
 - NACCESS 189
 - naive Bayes (NB) classifier 516
 - nanopore sequencing 509
 - National Biomedical Research Foundation (NBRF) 1
 - National Center for Biomedical Ontology (NCBO) 543
 - National Center for Biotechnology Information (NCBI) 1, 2, 9, 20, 27, 54, 55, 76, 254, 341, 444 *see* Entrez
 - annotation pipeline 144
 - ASN.1 (Abstract Syntax Notation) format 377
 - National Drug File (NDF) 543
 - National Institute of Standards and Technology (NIST) 343, 448, 458
 - National Institutes of Health (NIH) 19, 79
 - All of Us initiative 537, 546–7
 - natural selection 481–2, 493–7
 - NChannelSet 284
 - nearest centroid model 306
 - nearest neighbor free energy parameters 159
 - nearest neighbor model 157
 - Needleman–Wunsch algorithm 126, 260
 - negative binomial distribution 567
 - negative selection 493
 - NEIGHBOR program 266
 - neighbor-joining (NJ) 263
 - neighboring concept 21–3
 - Nematode Genome Annotation Assessment Project (nGASP) 128
 - NetMatchStar Cytoscape app 429
 - network inference 429–30
 - network smoothing 429
 - network visualization and analysis 425–6
 - neural networks 306
 - next-generation DNA sequencing (NGS) 133
 - neXtProt 205
 - NGL Viewer 374, 379
 - NMR-based compound identification 454–6
 - nmrML 441–3
 - NMRShiftDB 446, 448
 - NMRShiftDB2 448
 - no free lunch theorem 521
 - NOESY 365
 - nominal random variable 557
 - non-affine (or linear) gap penalty 52
 - non-metric multidimensional scaling (NMDS) 519, 520
 - non-negative matrix factorization (NMF) 291, 292, 296
 - non-parametric tests 578
 - normalized enrichment score (NES) 422
 - normalized unscaled standard error (NUSE)
 - boxplots 284–5
 - northern blot analysis 287
 - N-SCAN 136
 - nuclear magnetic resonance (NMR) 130, 364–6, 386, 400, 439
 - nuclear Overhauser effects (NOEs) 365
 - nucleotide scoring matrices 51
 - nucleotide sequence databases 3
 - nucleotide sequence flatfiles 3–11
 - RefSeq 10–11
 - header 4–6, 583–4
 - feature table 4, 6–9, 484–5
 - graphical interfaces 9
 - null hypothesis 570–71
 - testing 573–5
 - NUPACK 175
- o**
- Observational Health Data Sciences and Informatics (OHDSI) 544
 - Observational Medical Outcomes Partnership (OMOP) data model 544
 - observed heterozygosity 483
 - odds ratio 577
 - oligotyping 514
 - OMIM 33–7
 - OmniPath 414
 - OneDep 373, 374
 - Online Mendelian Inheritance in Man (OMIM) 33–7
 - OPAL 240

- Open Babel 441
 Open Biomedical Ontologies Consortium (OBO) 543
 OpenGL Shading Language (GLSL) 380
 Openmolecules.org 439
 OpenMS 352
 operational taxonomic unit (OTU) 514
 OPLS-DA 470
 OPSIN web server 439
 Orbitrap 317–320, 458, 459
 ORGANISM lines (in DDBJ/GenBank) 5
 organismal sequence databases 38–42
 OrthoDB 135
 Orthogonal Projection of Latent Structures–Discriminant Analysis (OPLS-DA) 468
 orthologs 45, 46, 255, 256
 over-dominance 493
 OXBench 230
 Oxford Nanopore MinION 509
- p**
- p* value 572–8
 P4 Medicine (Institute of Systems Biology) 547
 Pacific Biosciences (PacBio) RS II 509
 paired *t*-test 578, 579
 Paired-End reAd mergeR (PEAR) 524
 pairwise sequence alignments 4, 260
 pairwise sequentially Markovian coalescent (PSMC) model 488
 PAM (Point Accepted Mutation) 262
 PAM matrices 48–50
 PAM unit 49
 PANAV 374
 PANTHER 203
 paralogs 45, 255, 256
 paralogy 46
 parameter 558
 parametric tests 578
 parent ion (or formula) matching 459
 parsimony principle 264
 partial least-squares 306
 partial least-squares discriminant analysis (PLS-DA) 462, 467–70
 partition functions 159
 PARTS 163
 PASA 143, 144
 PASSEL 352
 PASTA 234, 235, 237–41
 Pathguide link directory 403, 414
 PathoLogic algorithm 401
 PathVisio 418, 420
 pathway analysis
 databases 402–31
 experiments and predictions 400–2
 pathway enrichment analysis 399, 419–21, 423, 424
 standard data formats 415–17
 topological analysis 471, 472
 visualization tools 417–25
 Pathway Commons 414, 417
 Pathway Tools software 405
 Patient-Centered Outcomes Research Institute data model 544
 PATRIC 341
 Paul ion trap (quadrupole ion trap) 318
 PAUP 265, 267
 PBS test 494
 PDBeFOLD 391
 PDLI 205
 PEAKS PTM 341
 Pearson correlation coefficient 195, 305, 560, 561
 Pearson correlation distance measure 292
 Penning trap (FT-ICR) 318
 PEPTIDE 354
 peptide mass fingerprinting 334–6
 peptide sequence tag searching 344
 peptide spectral matching 340–1, 345–7
 PeptideAtlas 352, 353
 PeptideAtlas SRM Experiment Library (PASSEL) 352
 PeptideProphet 345, 353
 pepXML 348
 Percolator 345
 PERMANOVA 521
 Perseus 347
 Personal Genome Project (PGP) 547
 Pfam 202–4, 230, 233, 234, 390
 Pfold 162, 178
 Phage_Finder 138
 Pharmacogenetics Knowledgebase (PharmGKB) 538
 PHAST 138
 PHASTER 138, 139
 PHDacc 190
 PHDsec 190, 191
 PhenoPred 541
 Phenotypic Quality Ontology (PATO) 543
 PheWAS Approach 545
 Phobius 197
 PhosphositePlus (PSP) 355
 photo-activated fragmentation 319
 Phred scores 509
 PHYLLIP 233, 265, 266

- phylogenetic analysis 251–273
 - data integration 269–72
 - determining the substitution model 262–3
 - early classification schemes 252–3
 - marker-based evolution studies 268–9
 - multiple sequence alignment and alignment editing 260–2
 - phylogenetic inference 263
 - phylogenetic placement 516
 - phylogenetic profile methods 402
 - sequences as molecular clocks 253–4
 - terminology 254–60
 - tree building 263–5
 - tree construction 260–7
 - tree visualization 265–7
- phylogenetic inference 263
- Phylogenetic Investigation of Communities by Reconstruction of Unobserved States (PICRUSt) 522–3
- phylogeography 270
- PhyloSift 526
- PhyML 265
- Phyre2 384
- PIFACE 211
- PileUp 260
- PIR 2, 12
- PIR+ 2
- PIRSF 203
- PISA 211
- PLINK 484, 490, 491
- PMF database search concept 334, 336
- point accepted mutation *see* PAM
- Poisson distribution 258, 567
- polymerase chain reaction (PCR) 506, 512, 513
- polynomial time dynamic programming algorithm 160
- PolyPhen 213
 - PolyPhen-2 213, 214, 541
- PolyPhobius 197, 198
- POODLE 200, 201
- population branch statistic (PBS) 494
- population genetics 481–99
 - admixture and ancestry estimation 489–93
 - allele frequencies and population variation 482–4
 - demographic history inference 485–9
 - display methods 484–5
 - evolutionary processes and genetic variation 481–2
 - theory 498
- positional frequency matrix (PFM) 119, 120
- positional probability matrix (PPM) 120, 121
- position-specific iterated BLAST *see* PSI-BLAST
- position-specific scoring matrices *see* PSSMs
- positive predictive value (PPV) 303
- positive selection 493, 494
- post-translational modifications (PTMs) 315, 324–5
 - databases 354
- POV-Ray 381
- Pplacer 516, 526
- Pprint 211
- PRANK 237, 238, 241
- PrDOS 200
- PrDOS-CNF 200, 201
- Precision 130
- precision medicine 537–8
- predictive network analysis 429
- PredictProtein 186, 190, 270
- PREFAB 230
- PRIDE 352
- principal component analysis (PCA) 286, 290–2, 294–6, 460–9, 484, 485, 490–3, 519, 564–5, 579
- principal coordinate analysis (PCoA) 519, 520
- PRINTS 203
- PRK 203
- ProbKnot 169, 170, 175
- process description (PD) diagrams 418
- PROCHECK 373, 387, 388
- PROCRUSTES 135
- Prodigal 123, 131
- PRODOM 203
- PROFacc 190
- ProfileAnalysis (Bruker) 458
- PROFsec 190
- Progenesis (Nonlinear Dynamics) 458
- progressive alignment 228
- Prokka 142
- Promethase 540
- ProPack 260
- Prophage Finder 138
- ProSight PTM 344
- PROSITE 202, 390
- PROTDIST 266
- protein characterization 315
- protein crystallography 364
- Protein Data Bank (PDB) 8, 9, 12, 22, 187, 190–2, 195, 196, 354
 - databases 450
 - format 369–71, 441
 - ID number 34
 - PDBe 373
- protein disorder 199
- Protein Explorer 379

- protein folds 368, 378
- Protein Information Resource (PIR) 1, 2
 PIR Protein Sequence Database 13
- protein interaction sites 210–12
- Protein Mutant Database 213
- Protein Research Foundation 12
- protein scoring matrices 46–52
- protein sequence databases 11–15
- protein sequences 185–215, 426
- protein structure 186–201, 363–73
 databases 373–7
 evaluation 386–8
 prediction 381–6
 primary 130, 367
 secondary 193–5, 367, 368
 tertiary 367–9
- Protein Workshop 374, 379
- Protein/Proteomics Databases 354–5
- ProteinProphet 353
- ProteoCloud 345
- proteofoms 329
- Proteome Discoverer software suite 346
- ProteomeXchange (PX) 352
- proteomics
 data repositories 352–4
 definition 315
 differential 355
 functional 355–6
 strategies 328–34
 structural 356
- proteome 315
- Proteomics Identifications database (PRIDE) 352
- Proteomics Standard Initiative (PSI) 348
see also under PSI
- proteomics XML formats 348, 352
- Proteopedia 377
- ProteoWizard 352
- Proteus 191, 192, 197, 382
- protXML 348
- PROVEAN 212–13
- provenance 404
- ProViz 246
- pseudocounts 120
- pseudoknot 160, 169
- PSI-BLAST 64–6, 191–3, 200, 203, 205–9, 211, 212, 231
- PSI-MI (Proteomics Standard Initiative–Molecular Interactions) 348, 407, 409, 417, 418
- PSIPRED 191, 195
- PSIVER 211
- PSSMs 64–6, 68, 69, 119, 121, 125, 135, 188, 192, 200, 202, 205, 206
- PubChem, 377, 444, 446–8
- PubMed 20–3, 444
- pulsed-field gel electrophoresis (PFGE) 268
- purifying selection 493
- PV (Protein Viewer) 374, 379
- PWMs 119, 125, 135
- PX Consortium 352
- PyMod 382
- PyMOL 196, 380, 382, 389
- pYST 354
- Python 427
- q**
- QC stat plot 285
- QIIME 513
- QModeler score 230
- QTOFs 458, 459
- quadratic discriminant analysis 306
- quadrupole mass analyzer 317, 318
- qualifiers 6
- QuanTest 231
- quantitative proteomics 329–32
- quaternary structure 367, 369
- quaternion methods 389
- r**
- R programming language 302, 427, 460
- Ramachandran plot 371–2, 380
- random forests 306, 470
- random variable 556
- Rapid Annotation using Subsystem Technology (RAST) 142
- RaptorX 384–5
- RaptorX Property 192, 195
- rarefaction curves 517, 518
- RasMol 378–9
- rate heterogeneity 258
- Ray Meta 524
- RCSB-PDB 373
- RDP 507
- Reactome 14, 405, 406, 417, 418, 446, 450, 471
- ReactomeFIViz Cytoscape app. 430
- read-mapping approaches 525
- READSEQ 266
- RECON 140
- REDUCE 388
- RefSeq 9–12, 27, 29–32, 55, 85, 87, 88, 113, 354
 RefSeq IDs 283
- relative abundance (evenness) of taxa 517
- relative log expression (RLE) boxplots 284
- relevance pairs model of retrieval 22

- Rebase 140
 - RepeatMasker 141
 - RepeatScout 140
 - ReProf 186, 190, 195
 - Research Collaboratory for Structural Bioinformatics (RCSB) 375
 - RetroPred 140
 - reverse phase liquid chromatography (RPLC) 344
 - reverse transcription polymerase chain reaction 282
 - RGASP 128, 133
 - ribbon diagram approach 378
 - Ribosomal Database Classifier 516
 - Ribosomal Database Project (RDP) 507, 513
 - Richard's Box 378
 - Richardson ribbon diagrams 378
 - RNA analysis programs 166
 - RNA pseudoknot 160
 - RNA-PUZZLES 177, 178
 - RNA secondary structure 157–9, 161–3
 - RNA-seq Genome Annotation Assessment Project (RGASP) 128, 133, 134
 - RNA sequences 155–78
 - RNA sequencing (RNA-seq) 133, 279–85
 - RNA tertiary structure 176–8
 - RNAalifold 176
 - RNABindRPlus 211
 - RNAML-compliant programs 166
 - RNAplex 175, 176
 - RNAstructure Web Server 156, 166–70
 - RNAup 175
 - Robetta 385
 - ROLLOFF 490
 - root-mean-square deviations (RMSDs) 22, 366
 - Rosetta 176, 385
 - ROSIE 385
 - rRNA Gene Finding 136–7
 - RxNORM 543
- S**
- SABmark 230
 - Sailfish 283
 - Salmon 283, 309
 - sample covariance 560
 - sample preparation 322, 324–5
 - sample size 573
 - sample variance 560
 - SAMtools 486
 - Sanger sequencing 507
 - SANN 192, 195
 - SATé 240
 - satisfaction of spatial restraints 382
 - Savitzky-Golay filtering 325
 - scatterplots 562, 564
 - SCONE 309
 - Scooby-Domain 205
 - scoring matrices 46–52
 - searching by content 118
 - searching by signal 118
 - SeaView 231, 232, 242, 243, 246
 - seed sequence 514
 - SEG 140
 - segment overlap score (SOV) 193, 199
 - selected reaction monitoring (SRM) 333
 - selective filtration 506
 - selex 233
 - sensitivity 128–30
 - SEQBOOT 266
 - Sequence Alignment/Map (SAM) format 82
 - sequence-function gap 185
 - sequence variants, effect of 212–14
 - Sequential Interval Motif Search (SIMS) 341
 - SEQUEST 341, 345–6
 - Sequin 373
 - serial analysis of gene expression (SAGE) 280
 - Seurat 309
 - SFCHECK 373
 - SFLD 203
 - Sfold 175, 177
 - SGP-2 132, 136
 - Shannon diversity index 514, 517
 - SHAPE 161, 170, 176
 - SHAPEIT2 program 486
 - ShapeKnots method 170
 - SHIFTX2 374
 - short-read Illumina sequence data 509
 - shotgun approach 80, 315, 322, 328–30, 333
 - shrunken nearest centroid model 306
 - SIFT 212–14, 541
 - SIFTER (Statistical Inference of Function Through Evolutionary Relationships) 270
 - SigmaFit (Bruker) 459
 - signal-to-noise ratio (SNR) 325
 - significance analysis of prognostic signatures (SAPS) 304–6
 - SILVA 513, 515
 - SIMCA 468
 - similarity 45
 - Simons Genome Diversity Project 485
 - Simple Viewer 374, 379
 - simpleaffy package 285

- SinaPlot 562
- single-cell RNA sequencing (scRNA-seq) 308–9
- single nucleotide polymorphisms (SNPs) 28, 29
- single-sequence methods, practical approach 163–70
- Using the Mfold Web Server 163–6
- singleton density score (SDS) 494
- SIRIUS 459
- SISTR (Salmonella In Silico Typing Resource) 268
- skewness 568
- SLAM 136
- Small Molecule Pathway Database (SMPDB) 446, 450
- SMART 202, 203, 548
- SMART On FHIR 548
- SMARTPCA 492
- SMART-Seq 308
- SMILES (Simplified Molecular Input Line Entry System) 440, 442
- Smith–Waterman algorithm 72, 75
- SMPDB 471, 473
- SNAP 354, 541
- SNAP2 213, 214
- SNPedia 540
- SNVs 268
- soft masking programs 140
- Soft Independent Modeling of Class Analogy (SIMCA) 468
- soft sweep 493, 494
- SOLID 203
- SomeNA 211
- SOURCE line (in DDBJ/GenBank) 5
- space-filling models 373
- Spearman’s rank correlation coefficient (Spearman’s correlation) 561
- specificity 128, 129
- specificity-determining positions (SDPs) 203
- spectral deconvolution 456–7
- spectral library searching 342–3
- spectral viewers 443–4
- SpectraST (PeptideAtlas) 343
- SPIDER3 193, 195
- spliced alignments 133
- spML 352
- spring-embedded algorithm 427
- squid library 233
- SSAP 229
- SSM 391
- SSpro5 192
- stable isotope probing 529
- STAMP 229, 521
- Stampy 133
- standard deviation (SD; StdDev) 560
- standard error of the mean (SE; SEM) 560
- standard normal distribution (Gaussian distribution) 566–7
- STAR 133
- STAR aligner 283
- statistical hypothesis testing 570–1
- statistical inference 569–70
- statistical power 575–6
- statistical significance 572–3
- StatQuest 345
- STRING Millennium 379
- STRIDE 189
- STRING 412–13
- StringTie 133
- structural bioinformatics 363
- Structural Classification of Proteins database (SCOP) 75, 203, 390
- Structural Protein Segments, scoring schemes for 198–9
- STRUCTURE 489–90
- Structure Data Format (SDF) 441
- Structure Prediction, *ab initio* 385–6
- Structured Query Language (SQL) 542
- STS (Sequence-Tagged Sites) 5
- Student’s *t*-test 297–8, 459, 573–6
- subcellular localization 208–210
- Substitutable Medical Apps and Reusable Technologies *see under* SMART
- substitution matrix 257
- SUPERFAMILY 203
- SuperPose 389
- support vector machines 121, 306, 470
- surface-induced dissociation (SID) 344
- surrogate variable analysis (SVA) 290
- SVM 211
- swarm clustering 514
- SWISS-MODEL server 382
- Swiss-PDB Viewer 380
- Swiss-Prot 65, 73, 203–5
- protein sequence database 2, 11, 527
- synonymous substitutions 257
- Systematized Nomenclature of Medicine–Clinical Terms (SNOMEDCT) 543
- Systems Biology Graphical Notation (SBGN) 416, 418
- Process Description (PD) 421, 422
- Systems Biology Markup Language (SBML) 417, 450

t

T92 model 262
 TANDEM 346–7
 tandem mass spectrometry (MS/MS, or MS2)
 319–24, 336–44, 458, 459
 TATA box 118
 taxonomy 251
 T-Coffee 235, 242
t-distributed stochastic neighbor embedding
 (*t*-SNE) plots 309
 telomeres 546
 threading (or fold recognition) 383–5
 Tide 346
 TIGRFAM 203
 time of flight (TOF) mass analyzer 317, 318
 TM-align 376, 389
 TMB-HUNT 197
 TMHMM 197
 TMSEG 197, 198
 TOCSY 365
 Top Hat filter 325
 top-down (intact protein) MS 344
 top-down proteomics (TDP) 329, 333
 TopHat2 133
 TopMatch 376
 TopPIC 344
 TopSearch 391
 torsion angles 369
 total ion chromatogram (TIC) 456
 Toxic Exposome Database (T3DB) 446, 450,
 451
 traits 252
 TraML 352
 transcripts per million (TPM) values 287–8
 translational bioinformatics 537–50
 electronic health records 542–4
 ethical, legal, and social implications of
 translational medicine (ELSI) 549–50
 human health genetics databases 538–40
 patient privacy 549–50
 prediction and characterization of impactful
 genetic variants 540–2
 precision medicine 544–9
 transmembrane proteins 195–9
 Transomics 133, 134
 TransPath (BioBase, Inc.) 450
 Trans-Proteomic Pipeline (TPP) software suite
 347
 Tree of Life 253–4
 tree space 264
 tree-building methods 258–9
 TreeTool 266

TREEVIEW 266, 267
 TrEMBL 2, 13
 Trembly 133, 134
 trimmed mean of *M* values (TMM) approach
 289–90
 Trinity 133
 triple quadrupole mass analyzer 319
 tRNA gene finding 136–7
t-test 521, 528, 573–5
 TurboFold 163, 170, 175, 176, 178
 TurboSEQUENT 346
 twilight zone 61
 TWINSKAN 136
 two-dimensional gel electrophoresis (2DGE)
 316
 two-sample *see* Student's *t*-test
 type I and II errors 571–2

u

Ubuntu 237
 UCHIME 513
 UCSC Genes track 84
 UCSC Genome Browser 70, 79–94, 96, 98,
 101, 110–3
 UCSC Table Browser 94–6, 108, 109, 112
 UK Biobank 537
 UK10K Project 485–6
 ultra-high-pressure liquid chromatography
 (UHPLC) 322
 ultra-high-throughput sequencing 279
 ultraviolet photodissociation (UVPD) 344
 UniFrac 518
 UniParc 13, 354
 UniProt 12, 340, 354–5, 444
 UniProt Archive 13
 UniProt Consortium 2, 13, 340, 354
 UniProt Knowledgebase *see* UniProtKB
 UniProt Reference Clusters (UniRef) 13, 354
 UniProtKB 2, 9, 12–14, 65, 68, 204–6, 208
 University of California, Santa Cruz (UCSC)
 79
 Genome Browser 145
 Unix/Linux machines 163
 untranslated regions (UTRs) 8
 Unweighted Pair Group Method with
 Arithmetic Mean (UPGMA) 263

v

validation of predictive models 306–8
 Variant Call Format (VCF) 81, 82, 486, 490,
 491, 540
 Variant Effect Predictor (VEP) 97

- Variants of unknown significance (VUS) 540
Vector Alignment Search Tool (VAST) 21, 22, 32, 34, 377
ViennaRNA Package 162, 175
violin plot 562, 563, 568
virtual proteomes 315
Viterbi algorithm 126
Volume, Area, Dihedral Angle Reporter (VADAR) 387–8
Voom 298
- W**
- Warburg effect 471
Web Apollo 145
Web Ontology Language (OWL) 416
WebGraphics Library (WebGL) 379–80
WebMol 379
weighted correlation network analysis (WGCNA) 430
Welch ANOVA 578, 579
Welch unequal variance t-test 579
Wellcome Trust Sanger Institute (WTSI) 80
WHAT_CHECK 373
WikiPathways 446, 450, 471
Wilcoxon-Mann-Whitney test 578
Wilson plots 373
- WineScreener 456
wireframe models 378
Worldwide Protein Data Bank (wwPDB) 373, 374, 567
WormBase 11, 341
- X**
- X! Hunter 343, 353
X! Tandem 341, 346–7, 353
XCMS 458–60, 529
XCMS-Plus (Sciex) 458
XDrawChem 443
xenologs 255, 256
XML 347, 441
XP-EHH test 494, 497
X-ray crystallography 130, 364, 365
- Y**
- Yeast Metabolome Database (YMDB) 446, 450, 451, 471
yeast two-hybrid 401
- Z**
- Zebrafish Information Network (ZFIN) 40, 41
Zebrafish Model Organism Database 40

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley's ebook EULA.