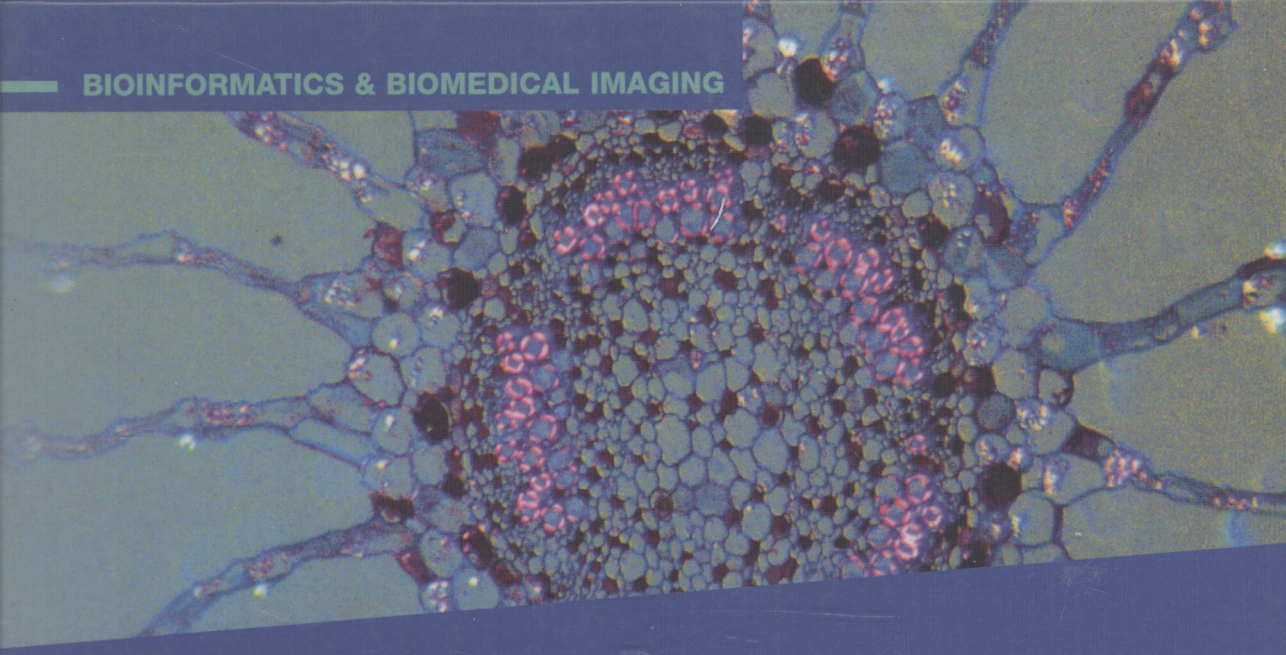
BIOINFORMATICS & BIOMEDICAL IMAGING



# Biological Database Modeling

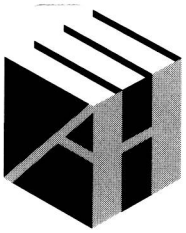
Jake Chen • Amandeep S. Sidhu
*editors*

# Biological Database Modeling

Jake Chen
Amandeep S. Sidhu

*Editors*





ARTECH
HOUSE
BOSTON | LONDON
artechhouse.com

**Library of Congress Cataloging-in-Publication Data**
A catalog record for this book is available from the U.S. Library of Congress.

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

ISBN-13: 978-1-59693-258-6

**Cover design by Igor Valdman**


**685 Canton Street**
**Norwood, MA 02062**



10 9 8 7 6 5 4 3 2 1

# Biological Database Modeling

Artech House Bioinformatics & Biomedical Imaging
Stephen T. C. Wong and Guang-Zhong Yang, Series Editors

*Advanced Methods and Tools for ECG Data Analysis,* Gari D. Clifford, Francisco Azuaje, and Patrick E. McSharry, editors

*Biomolecular Computation for Bionanotechnology,* Jian-Qin Liu and Katsunori Shimohara

*Electrotherapeutic Devices: Principles, Design, and Applications,* George D. O'Clock

*Intelligent Systems Modeling and Decision Support in Bioengineering,* Mahdi Mahfouf

*Life Science Automation Fundamentals and Applications,* Mingjun Zhang, Bradley Nelson, and Robin Felder, editors

*Matching Pursuit and Unification in EEG Analysis,* Piotr Durka

*Microfluidics for Biotechnology,* Jean Berthier and Pascal Silberzan

*Systems Bioinformatics: An Engineering Case-Based Approach,* Gil Alterovitz and Marco F. Ramoni, editors

*Text Mining for Biology and Biomedicine,* Sophia Ananiadou and John McNaught

List of Contributors


Chapter 1 Amandeep S. Sidhu,
*Curtin University of Technology, Perth, Australia*
Jake Chen,
*Indiana University School of Informatics, Indianapolis, Indiana, United States*

Chapter 2 Viroj Wiwanitkit,
*Chulalongkorn University, Bankok, Thailand*

Chapter 3 Ramez Elmasri, Feng Ji, and Jack Fu,
*University of Texas at Arlington, Arlington, Texas, United States*

Chapter 4 Viroj Wiwanitkit,
*Chulalongkorn University, Bankok, Thailand*

Chapter 5 Amandeep S. Sidhu, Tharam S. Dillon, and Elizabeth Chang,
*Curtin University of Technology, Perth, Australia*

Chapter 6 Cornelia Hedeler and Paolo Missier,
*University of Manchester, Manchester, United Kingdom*

Chapter 7 Greg Butler, Wendy Ding, John Longo, Jack Min, Nick O'Toole, Sindhu Pillai, Ronghua Shu, Jian Sun, Yan Yang, Qing Xie, Regis-Olivier Benech, Aleks Spurmanis, Peter Ulycznyj, Justin Powlowski, Reg Storms, and Adrian Tsang,
*Concordia University, Canada*

Chapter 8 Willy A. Valdivia-Granda,
*Orion Integrated Biosciences, Inc., New York, United States*
Christopher Dwan,
*BioTeam, Inc., Cambridge, Massachusetts, United States*

Chapter 9 Zhong Yan and Jake Chen,
*Indiana University School of Informatics, Indianapolis, Indiana, United States*
Josh Heyen, Lee W. Ott, Maureen A. Harrington, and Mark G. Goebl,
*Indiana University School of Medicine and the Walther Cancer Institute, Indianapolis, Indiana, United States*
Cary Woods,
*Ball State University, Muncie, Indiana, United States*

Chapter 10 Karthik Raman, Yeturu Kalidas, and Nagasuma Chandra,
*Supercomputer Education and Research Centre and Bioinformatics Centre, Indian Institute of Science, India*

Chapter 11 Preeti Malik, Tammy Chan, Jody Vandergriff, Jennifer Weisman, Joseph DeRisi, and Rahul Singh,
*San Francisco State University, San Francisco, California, United States*

# Preface

Database management systems (DBMS) are designed to manage large and complex data sets. In the past several decades, advances in computing hardware and software and the need to handle rapidly accumulating data archived in digital media have led to significant progress in DBMS research and development. DBMS have grown from simple software programs that handled flat files on mainframe computers, which were prohibitively expensive to all but a few prestigious institutions, into today's popular form of specialized software platforms underpinning wide ranges of tasks, which include business transactions, Web searches, inventory management, financial forecasts, multimedia development, mobile networks, pervasive computing, and scientific knowledge discovery. Technologies of DBMS have also become increasingly sophisticated, diverging from generic relational DBMS into object-relational DBMS, object-oriented DBMS, in-memory DBMS, semantic Webs data store, and specialized scientific DBMS. Given the sustained exponential data growth rate brought forth by continued adoption of computing in major industries and new inventions of personal digital devices, one can safely predict that DBMS development will continue to thrive in the next millennium.

In this book, we want to share with our readers some fresh research perspectives of post-genome biology data management, a fast-growing area at the intersection of life sciences and scientific DBMS domains. Efficient experimental techniques, primarily DNA sequencing, microarrays, protein mass spectrometers, and nanotechnology instruments, have been riding the wave of the digital revolution in the recent 20 years, leading to an influx of high-throughput biological data. This information overload in biology has created new post-genome biology studies such as genomics, functional genomics, proteomics, and metabolomics—collectively known as "omics" sciences in biology. While most experimental biologists are still making the transition from one-gene-at-a-time type of studies to the high-throughput data analysis mindset, many leaders of the field have already begun exploring new research and industrial application opportunities. For example, managing and interpreting massive omics data prelude ultimate systems biology studies, in which one may analyze disparate forms of biological data and uncover coordinated functions of the underlying biological systems at the molecular and cellular signalling network level. On the practical side, understanding diverse intricate interplays between environmental stimuli and genetic predisposition through omics evidence can help pharmaceutical scientists design drugs that target human proteins with high therapeutic values and low toxicological profiles. With data management tools to handle terabytes of omics data already released in the public domain, the promise of post-genome biology looms large.

Compared with data from general business application domains, omics data has many unique characteristics that make them challenging to manage. Examples of these data management challenges are:

1. Omics data tends to have more complex and more fast-evolving data structures than business data. Biological data representation often depends on scientific application scenarios. For example, biological sequences such as DNA and proteins can be either represented as simple character strings or connected nodes in three-dimensional spatial vectors. Data representation is an essential first step.
2. Omics data is more likely to come from more heterogeneously distributed locations than business data. To study systems biology, a bioinformatics researcher may routinely download genome data from the Genome Database Center at the University of California, Santa Cruz, collect literature abstracts from the PubMed database at the National Library of Medicine in Maryland, collect proteome information from the Swiss-Prot database in Switzerland, and collect pathway data from the KEGG database in Japan. Data integration has to be carefully planned and executed.
3. Omics data tends to reflect the general features of scientific experimental data: high-volume, noisy, formatted inconsistently, incomplete, and often semantically incompatible with one another. In contrast, data collected from business transactions tends to contain far fewer errors, is often more accurate, and shows more consistencies in data formats/coverage. Meticulous data preprocessing before knowledge discovery are required.
4. Omics data also lags behind business data in standard development. For example, Gene Ontology (GO) as a standard to control vocabularies for genes was not around until a decade ago, whereas standards such as industrial product categories have been around for decades. The ontology standards and naming standards for pathway biology are still under development. This makes it difficult to perform mega collaboration, in which cross-validation of results and knowledge sharing are both essential.

Despite all the challenges, modeling and managing biological data represent significant discovery opportunities in the next several decades. The human genome data bears the ultimate solutions of expanding the several thousand traditional molecular drug targets into tens of thousands genome drug targets; molecular profiling information, based on individuals using either the microarrays or the proteomics platform, promises new types of molecular diagnostics and personalized medicine. As new applications of massive biological data emerge, there will be an increasing need to address data management research issues in biology.

In this compiled volume, we present to our readers a comprehensive view of how to model the structure and semantics of biological data from public literature databases, high-throughput genomics, gene expression profiling, proteomics, and chemical compound screening projects. The idea of compiling this book, which we found to be unique, stems from the editors' past independent work in bioinformatics and biological data management. While topics in this area are diverse and interdisciplinary, we focused on a theme for this book—that is, how to model and manage

omics biological data in databases. By promoting this theme for the past decade among ourselves and the contributing authors of this book, we have contributed to solving complex biological problems and taking biological database management problems to the next level. We hope our readers can extract similar insights by using this book as a reference for future related activities.

There are 11 chapters presented in this book. Individual chapters have been written by selected accomplished research teams active in the research of respective topics. Each chapter covers an important aspect of the fast-growing topic of biological database modeling concepts. Each chapter also addresses its topic with varying degrees of balance between computational data modeling theories and real-world applications.

In Chapters 1 through 5, we introduce basic biological database concepts and general data representation practices essential to post-genome biology. First, biological data management concepts are introduced (Chapter 1) and major public database efforts in omics and systems biology studies are summarized (Chapter 2). Then, biomedical data modeling techniques are introduced (Chapter 3). Next, Gene Ontology as an established basic set of controlled vocabulary in genome database annotations is described (Chapter 4). Finally, the latest research on protein ontology and the use of related semantic webs technologies are presented to enable readers to make the connection between emerging biological data collection and integration trends (Chapter 5).

In Chapters 6 through 9, we examine in detail how to develop data management techniques to process and analyze high-throughput biological data through case studies. First, quality control techniques to reduce variations during experimental data collection steps are described (Chapter 6). Then, biological sequence management experience for a fungi genomics project is discussed (Chapter 7). Next, data management and data integration methods for microarray-based functional genomics studies are investigated (Chapter 8). Finally, data management challenges and opportunities for mass spectrometry based expression proteomics are presented (Chapter 9).

In Chapters 10 and 11, we delve into the practical aspect, demonstrating how to apply biological data management for drug discoveries. First, fundamental drug discovery concepts based on macromolecular structural modeling are introduced (Chapter 10); then, a data management software system that implements high-throughput drug compound screenings is discussed (Chapter 11) to conclude the book.

We hope this book will become a useful resource for bioinformatics graduate students, researchers, and practitioners interested in managing post-genome biological data. By studying the techniques and software applications described in this book, we hope that bioinformatics students will use the book material as a guide to acquire basic concepts and theories of post-genome biological data management, bioinformatics practitioners will find valuable lessons for building future similar biological data management systems, and researchers will find rewarding research data management questions to address in the years to come.

# Acknowledgments

We wish to thank all of the authors for sharing their insightful knowledge and making excellent contributions to this book based on their active research in biological data management. This book would not have been completed without the tremendous efforts, held to the highest standards, of all the authors, each of whom spent numerous hours over many drafts in preparing, collating, and revising their writings over the past 2 years. During the publishing process, many colleagues also helped and they deserve our whole-hearted appreciation. They are: David Wong from Indiana University, who provided legal advice for the contract agreement; Dr. Zongmin Ma from Northwestern University of China, who provided assistance in the initial conceptualization and execution of the book publishing process; Susan Lagerstrom-Fife from Springer Science+Business Media, Inc., who guided us from a publisher's perspective while we explored various publication options; Wayne Yuhasz from Artech House Publishers, whose persistence and dedication to timely assistance finally won us over in making our book part of the Artech House Bioinformatics & Biomedical Imaging Series; and Barbara Lovenvirth from Artech House Publishers, who assisted us throughout the final publication process.

We also want to thank the general support from our home institutions: Indiana University School of Informatics at Indianapolis, Indiana; Purdue University School of Science, Department of Computer and Information Sciences at Indianapolis, Indiana; and Curtin University of Technology, Perth, Australia.

Last, but not least, Jake Chen wishes to thank his family, including his wife, Mabel Liu, for her unbounded love and support in assuming additional responsibilities so that the book project could be completed.

*Jake Chen*
*Indianapolis, Indiana*
*Amandeep S. Sidhu*
*Perth, Australia*
*Editors*
*October 2007*

# Contents

CHAPTER 11

CHAPTER 1

# Introduction to Data Modeling

Amandeep S. Sidhu and Jake Chen

Scientific data is often scattered among heterogeneous data repositories. Exploring data across multiple data repositories requires the ability to understand and correlate their structures (schemas). Such correlations need to address the diversity of views of the scientific domain represented by different data repositories as well as the diversity of data modeling languages used for expressing these views. In this chapter, we introduce the concepts of *data modeling* and discuss its application to *biological databases*.

## 1.1 Generic Modern Markup Languages

Modern markup languages, such as Standard Generalized Markup Language (SGML) [1] and eXtensible Markup Language (XML) [2], which were initially conceived for modeling texts, are now receiving increasing attention as formalisms for data and knowledge modeling. XML is currently establishing itself as a successor of HyperText Markup Language (HTML) for a better modeling of texts as well as of other kinds of data. There are several reasons for this evolution. Even though multiple databases may cover the same data, their focus might be different. Modern markup languages such as SGML and XML are generic in that:

- They serve to specify the semantic structure, not the layout, of documents or data items.
- They make it possible to freely specify application-dependent document or data structures.

In the following, the term "data" refers also, but not exclusively, to text data. Thus, a data item may consist of: (1) text only (such data items are also known as human-readable documents); (2) nontext only (such data items are also known as data-oriented documents); or (3) both (such data items are also known as mixed-model documents). In the terminology of generic markup languages, data items are called documents. In the following, the term "data item" is used in lieu of "document" for stressing that not only (structured) texts are meant, but more generally (structured) data of any kind.

Widespread specific markup languages such as PostScript or Rich Text Format (RTF), whose conceptual roots go back to the 1970s, serve to specify the layout of data items. Here, layout is not exclusively meant as the appearance of a data item when printed on paper, but more generally as any kind of presentation of a data item to human perception. Examples of such an extended notion of layout include the formats of data items as they are displayed on a terminal screen, rendered in the script on an output device, or presented by any other means on any device.

The family of generic markup languages started in the late 1980s with the conception of its first specimen, SGML. The purpose of a generic markup language is to specify the semantic—or logical—structure of data items, not their layout. In the following, the term "presentation" is reserved to refer to the layout of a data item in the extended sense above, while the term "representation" refers to how semantics is conveyed through structural elements of the underlying data modeling formalism.

The distinction between layout and structure is important, for a layout format is device or system dependent, whereas a semantic structure should not be. It is desirable that the semantic structure of data items be specified independently of any layout. This ensures both:

- Independence of data modeling from data usage;
- Independence of data modeling from presentation devices.

The first property, data independence from usage, is important because data is rarely used in a single manner only. The second property, data independence from presentation devices, is important for several reasons. To begin with, different kinds of presentation devices require different layouts. For example, a structurally complex data item is likely not to be displayed using identical layouts on standard size screens and on small screens like those of cellular phones. Also, such devices are likely to become technically obsolete sooner than data. Moreover, a presentation format does not necessarily fully convey data semantics. For instance, it is common practice to rely on printed text layout for conveying semantic structure when using text processing systems or the markup language HTML. This practice often leads to semantic losses, especially when files are transferred from one text processing system to another, because the layout of the one system cannot always be faithfully mapped into that of the other system.

In order to specify layouts for classes of documents specified in a generic markup language, so-called style-sheet languages are used in addition. These languages basically allow the definition of layouts for those structural elements specified with the markup language. Such definitions do not have to be unique, thus ensuring the desired independence of the data from their presentations in various contexts.

Generic markup languages (like the XML family of languages) do not impose any predefined structure, nor any predefined names for the structural elements occurring in data items. Structure and names can be freely chosen, hence the denomination of generic markup language. Thus, using generic markup languages it is possible to faithfully model the structure of data items needed in applications and to name the structural elements of a chosen structure in a way that is natural in the application context.