PARALLEL COMPUTING FOR BIOINFORMATICS AND COMPUTATIONAL BIOLOGY

MODELS, ENABLING TECHNOLOGIES, AND CASE STUDIES

Edited by

Albert Y. Zomaya
The University of Sydney, Australia

WILEY-
INTERSCIENCE
JOHN WILEY & SONS, INC

PARALLEL COMPUTING FOR BIOINFORMATICS AND COMPUTATIONAL BIOLOGY

PARALLEL COMPUTING FOR BIOINFORMATICS AND COMPUTATIONAL BIOLOGY

MODELS, ENABLING TECHNOLOGIES, AND CASE STUDIES

Edited by

Albert Y. Zomaya
The University of Sydney, Australia

WILEY-
INTERSCIENCE
JOHN WILEY & SONS, INC

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. The publisher is not engaged in rendering professional services, and you should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic format. For more information about Wiley products, visit our web site at www.Wiley.com.

Library of Congress Cataloging-in-Publication Data is available.

ISBN-13 978-0-471-71848-2
ISBN-10 0-471-71848-3

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

To our families for their help, support, and patience.
Albert Zomaya

CONTENTS

PREFACE

Bioinformatics and Computational Biology are fields that requires skills from a variety of fields to enable the gathering, storing, handling, analyzing, interpreting, and spreading of biological information. It requires the use of high-performance computers and innovative software tools to manage enormous quantities of genomic and proteomic data. It also involves the development and application of innovative algorithmic techniques necessary for the analysis, interpretation, and prediction of data to provide insight into the design and validation of experiments for the life sciences.

Most of the above functionalities require the capabilities that are beyond those of a desktop machine and can only be found in a supercomputer. This is especially true now with the rapid increase of the amounts of data generated on a daily basis. Therefore, high-performance computing systems are expected to play an increased role in assisting life scientists in exploring possibilities that were impossible in the past. In return, the variety and richness of problems offered by bioinformatics and computational biology open up new vistas for computer scientists, which could keep them occupied for the next 50 years.

The book is based on a number of standalone chapters that seek to provide an opportunity for researchers to explore the rich and complex subjects of bioinformatics and computational biology and the use of parallel computing techniques and technologies (parallel computing, distributed computing, grid computing, etc.) in solving problems in these dynamic disciplines.

However, as with any new discipline, related applications should be designed and implemented in such a way that enables users to depend on the application availability and results. This book aims to highlight some of the important applications in bioinformatics and computational biology and to identify how parallel computing can be used to better implement these applications.

BOOK OVERVIEW

This is the first book that deals with the topic of parallel computing and its use to drive applications in bioinformatics and computational biology in such a comprehensive manner. The material included in this book was carefully chosen for quality and relevance. This book also provides a mixture of algorithmics, experiments, and simulations, which provide not only qualitative but also quantitative insights into the rich field of bioinformatics and computational biology.

This book is intended to be a repository of case studies that deal with a variety of difficult problems and how parallel computing was used to produce better results in a more efficient manner. It is hoped that this book will generate more interest in developing parallel solutions to wider life sciences applications. This should enable researchers to deal with more complex applications and with larger and richer data sets.

Although the material in this book spans a number of bioinformatics and computational biology applications, the material is written in a way that makes the book self-contained so that the reader does not have to consult with external material. This book offers (in a single volume) a comprehensive coverage of a range of bioinformatics and computational biology applications and how they can be parallelized to improve their performance and lead to faster rates of computations.

This book is intended for researchers, educators, students, and practitioners in the fields of bioinformatics, computational biology, and computer science, who are interested in using high-performance computing to target applications in the life sciences. This book can also be used as a reference for graduate level courses. This book is divided into five parts: algorithms and models, sequence analysis and microarrays, phylogenetics, protein folding, and platforms and enabling techniques. In what follows is a brief précis of the chapters included.

Chapter 1, after an introduction to genes and genomes, describes several efficient parallel algorithms that efficiently solve applications in computational biology. An evolutionary approach to computational biology is presented based first on the search space, which is the set of all possible solutions. The second factor used for the formulation of an optimization problem is the determination of a fitness function that measures how good a particular answer is. Finally, a significant deviation from the standard parallel solution to genetic parallel algorithms approach theory is pointed out by arguing that parallel computational biology is an important sub-discipline that merits significant research attention and that combining different solution paradigms is worth implementing.

Chapter 2 introduces an approach to simulating the molecular evolution of human immunodeficiency virus type 1 (HIV-1) that uses an individual virus-based model of viral infection of a single patient. Numerical methods, including Monte Carlo, are used to realistically simulate viral mutation, recombination, replication, infection, and selection by cell-surface receptor molecules and neutralizing antibodies. The stochastic nature of various events being simulated, such as mutation and recombination, requires that simulations be replicated to account for stochastic variation. In addition, because of the high level of realism, simulations may take a long time to run, and so replicate simulations are preferably run in parallel. The applications of the message-passing interface and the scalable parallel random number generator interface to this problem are described.

To analyze a biological system it is necessary to find out new mathematical models allowing to explain the evolution of the system in a dynamic context or to dread doing of a simple manner the complex situations where the human experience overtakes the mathematical reasoning. Computers have been used since the 1940s to simulate the kinetics of biochemical reactions. Using a pathway structure and a kinetic scheme, the time of reaction and the admissible steady states can be computed. These are discussed in Chapter 3.

A cell is an incredibly complex object as are the dynamical processes that take place within the cell. In spite of this complexity we can hope to understand the dynamics of a cell by building up a set of models and simulation approaches that can lock together in a modular fashion. The focus of Chapter 4 is on how stochasticity manifests itself in cellular processes and how this stochasticity can be modeled, simulated, and visualized. In particular, this chapter addresses the issues of how to simulate stochastic chemical kinetics in both temporal and spatial settings using both sequential parallel computing environments. The models for these simulations are associated with genetic regulation within a single cell but this work also considers colonies of cells.

The purpose of Chapter 5 is to survey some recent developments in the application of parallel and high-performance computation in simulating the diffusion process in the human brain and in modeling the deformation of the human brain. Computational neuroscience is a branch of biomedical science and engineering in which sophisticated high-performance computing techniques can make a huge difference in extracting brain anatomical information non-invasively and in assisting minimal invasive neurosurgical interventions. This chapter demonstrates that there are lots of potential opportunities for computational scientists to work with biomedical scientists to develop high-performance computing tools for biomedical applications.

In Chapter 6, the authors first introduce several basic concepts of molecular biology. This is then followed by a definition of the global and local sequence alignment problems and the exact algorithms used to solve them which are normally based on dynamic programming to solve them. The authors also present several heuristics that can be used to solve the local alignment problem. The chapter concludes with a description of some parallel algorithms that can be used to solve the alignment problems in shorter time.

Chapter 7 presents a hybrid parallel system based on commodity components to gain supercomputer power at low cost. The architecture is built around a coarse-grained PC cluster linked to a high-speed network and fine-grained parallel processor arrays connected to each node. Identifying applications that profit from this kind of computing power is crucial to justify the use of such a system. This chapter presents an approach to high-performance protein database scanning with hybrid computing. To derive an efficient mapping onto this architecture, we have designed instruction systolic array implementations for the Smith–Waterman and Viterbi algorithm. This results in a database scanning implementation with significant run-time savings.

Chapter 8 presents a parallel version of ClustalW for multiple sequence alignment. The algorithm is implemented using the message-passing interface (MPI), a platform for implementing parallel algorithms on a distributed shared memory model. This chapter presents a tutorial introduction to the ClustalW algorithm. First, the authors discuss the dynamic programming algorithm for pairwise sequence alignment. Then this is followed by a discussion of the neighbor-joining method of Seitou and Nei for constructing a phylogenetic tree using the pairwise distances. Finally, the authors present the progressive sequence alignment step based on this phylogenetic tree.

They discuss their strategy for parallelizing the ClustalW algorithm next and provide detailed results for their implementation and analyze the results extensively.

Chapter 9 examines several high-performance versions of BLAST, which is one of the most widely used search tools for screening large sequence databases. Even though BLAST is very efficient in practice, the growing size of sequence databases has created a demand for even more powerful versions of BLAST for use on multiprocessors and clusters. This chapter briefly reviews the basic BLAST algorithm, then describe and analyze several parallel versions of BLAST designed for high performance.

The purpose of pairwise alignment is to extract the sequences that are similar (homologous) to a given input sequence from a database of target sequences. While CPU architectures are struggling to show increased performance, the volume of biological data is greatly accelerating. For example, GenBank, a public database of DNA, RNA, and protein sequence information, is doubling every 6 months. Parallel algorithms for analyzing DNA and protein sequences are becoming increasingly important as sequence data continue to grow. Novel parallel architectures are also being proposed to deal with the growth in computational complexity. Chapter 10 reviews the parallel software and hardware implementations of local sequence alignment techniques. These include various implementations of Smith–Waterman algorithm, FASTA, BLAST, HMMER, and ClustalW.

DNA microarrays provide the technology needed to study gene expression. This technology facilitates large-scale surveys of gene expression in which transcript levels can be determined for thousands of genes simultaneously. These experiments generate an immense quantity of data. Investigators need computational methods to analyze this data to gain an understanding of the phenomena the data represent. Chapter 11 presents two advanced methods for analyzing gene expression data that go beyond standard techniques but require the use of parallel computing. The first method provides for the assessment of the codetermination of gene transcriptional states from large-scale simultaneous gene expression measurements with cDNA microarrays. The parallel implementation exploits the inherent parallelism exhibited in the codetermination methodology that the authors apply. The second method involves classification using cDNA microarrays. The goal is to perform classification based on different expression patterns such as cancer classification. The authors present an efficient parallel implementation of the σ-classifier where the computational work is distributed among available system processors.

As more research centers embark on sequencing new genomes, the problem of DNA fragment assembly for shotgun sequencing is growing in importance and complexity. Accurate and fast assembly is a crucial part of any sequencing project and many algorithms have been developed to tackle it. As the DNA fragment assembly problem is NP-hard, exact solutions are very difficult to obtain. Various heuristics, including genetic algorithms, were designed for solving the fragment assembly problem. Although the sequential genetic algorithm has given good results, it is unable to sequence very large DNA molecules. In Chapter 12, the authors present a distributed genetic algorithm that surmounts that problem. They show how the distributed genetic algorithm can tackle problem instances that are 77K base pairs long accurately.

DNA microarrays allow the simultaneous measurement of the expression level of thousands of genes. This is a great challenge for biologists who see in this new technology the opportunity to discover interactions between genes. The main drawback is that data generated with such experiments is so large that very efficient knowledge discovery methods have to be developed. This is the aim of Chapter 13. The authors propose to study microarray data by using association rules via a combinatorial optimization approach. A cooperative method, based on an evolutionary algorithm, is proposed and several models are tested and compared.

Chapter 14 provides a brief review of phylogenetics and provides an introduction to the maximum likelihood method (one of the most popular techniques used in phylogeney) and describes the abstract computational problems which arise at the computation of the likelihood score for one single-tree topology. This is followed by state-of-the-art description of sequential and parallel maximum likelihood programs. This chapter also explains the maximum likelihood program development cycle and describes algorithmic as well as technical enhancements of RAxMLIII. The chapter concludes by addressing promising technical and algorithmic developments and solutions which could enable the computation of larger and more accurate trees in the near future.

Phylogenetic analysis is a routine task in biological research. Chapter 15 discusses the different factors that influence the performance of parallel implementations. Using the example of parameter estimation in the TREE-PUZZLE program, the authors analyze the performance and speedup of different scheduling algorithms on two different kinds of workstation clusters, which are the most abundant parallel platform in biological research. To that end different parts of the TREE-PUZZLE program with diverse parallel complexity are examined and the impact of their characteristics is discussed. In addition, an extended parallelization for the parameter estimation part of the program is introduced.

Phylogenetic trees are extremely useful in many areas of biology and medicine, and one of the primary tools for understanding evolution. Unfortunately, for a given set of organisms, the number of possible evolutionary trees is exponential. Many phylogenetic algorithms exist, but the most popular approaches attempt to solve difficult optimization problems such as maximum parsimony (NP-hard) or maximum likelihood (conjectured to be NP-hard). Chapter 16 surveys the state-of-the-art in phylogenetic algorithms for reconstructing maximum parsimony trees. Each new algorithmic development attempts to get us closer to reconstructing the “Tree of Life,” the holy grail of phylogenetics. Thus, this chapter concludes with a list of research questions that must be addressed to reconstruct extremely large-scale phylogenies such as the “Tree of Life.”

A highly parallel replica exchange molecular dynamics (REMD) method and its application in protein folding and protein structure prediction are described in Chapter 17. The REMD method couples molecular dynamics trajectories with a temperature exchange Monte Carlo process for efficient sampling of the conformational space. Two sample protein systems, one α-helix and one β-hairpin, are used to demonstrate the power of the algorithm. Up to 64 replicas of solvated protein systems are simulated in parallel over a wide range of temperatures. Very high efficiency ($>98\%$) can be achieved with this embarrassingly parallel algorithm. The simulation results show that the combined trajectories in temperature and configurational space allow a replica to overcome free energy barriers present at low temperatures. These large-scale simulations also reveal detailed results on folding mechanisms, intermediate-state structures, thermodynamic properties, and the temperature dependencies for both protein systems. Furthermore, the extensive data from REMD simulations are used to assess the various solvation models and force fields, which provide insights to the fix of the problems and further improvement of the models. Finally, the usage of the REMD method in protein structure refinement is also discussed.

Chapter 18 deals with a method known as threading which uses information about already known protein structures stored in databases. The authors present the point of view of a computer scientist with particular interests in combinatorial optimization problems. They focus on the computational aspects of finding the optimal sequence-to-structure alignment referred as protein-threading problem (PTP). A formal definition of the PTP is given, and several mixed integer models are presented in a unified framework, analyzed, and compared. Different divide-and-conquer strategies are also described. They reduce the time needed to solve the master problem by solving auxiliary sub-problems of a moderate size. One section is particularly dedicated to a parallel implementation of such a technique, which happened to be efficient even in a sequential implementation. The results in this chapter demonstrate that a careful combination of modeling, decomposing, and a parallel implementation leads to solving PTP real-life instances of tremendous size in a reasonable amount of time.

In Chapter 19, the authors report results of a parallel modified fast messy GA (fmGA), which is found to be quite “good” at finding semi-optimal protein structure prediction solutions in a reasonable time. They focus on modifications to this EA called the fmGA, extensions to the multiobjective implementation of the fmGA (MOfmGA), constraint satisfaction via Ramachandran plots, identifying secondary protein structures, a farming model for the parallel fmGA (pfmGA), and fitness function approximation techniques. These techniques reflect marked improvement over previous GA applications for protein structure determination. Problem definition, protein model representation, mapping to algorithm domain, tool selection modifications, and conducted experiments are discussed.

Over the last few years Grid Computing has generated considerable interest among researchers, scientific institutions, research centers, universities, governments, funding bodies, and others. Grid technology can be used for many applications in the life sciences that require high computational power, data-intensive processing, storage management, and resource sharing. Chapter 20 reviews the current worldwide activities in Grid Computing as used to drive applications in bioinformatics and the health sciences. The chapter attempts to categorize grid activities by region and by the nature of the application. The review is by no means exhaustive and it is only meant to give the reader an appreciation that current applications that are benefiting from grid deployment and could also provide the thrust for future developments.

Chapter 21 discusses parallel algorithms for bioinformatics in the context of the Cray MTA architecture. This chapter shows how several bioinformatics algorithms can be implemented on this machine and develops an entirely new algorithm for DNA sequencing with very long reads that was developed with the MTA as target architecture. The chapter provides the insights that the authors gained by using the MTA architecture and shows that parallel algorithms may be implemented on this machine with a minimum of rewriting or reorganization. Finetuning of code requires only a basic understanding of the architecture and of the behavior of the tagged memory. The issues of data reorganization, partitioning, scheduling, mapping, and so on, which are central to conventional parallel processors, are nonexistent on this machine. The MTA is thus the ideal machine for a rapidly advancing field like bioinformatics, where algorithm development and coding must charge ahead in tandem.

Many computational chemists requiring significant and relatively flexible resources have turned to parallel clusters to solve increasingly complex problems. Evolving hardware technology and grid resources present new opportunities for chemistry and biology, yet introduce new complexity related to grid, web, and computational difficulties. Chapter 22 describes the author's experience in using the GAMESS quantum chemistry program on clusters, and their utilization of evolving portal, grid, and workflow technologies to solve problems that would not be practical on individual machines.

Chapter 23 sets forth the challenges faced by grid computing and discusses the nature of applications that can be grid-enabled. It introduces a framework that can be used to develop grid-enabled bioinformatics applications and provide examples that show how this can be achieved. The author argues that a software development framework for bioinformatics can only receive acceptance if all the complexity can be hidden away from the scientists. That is why such environments need to have sophisticated graphical user interfaces that enable the easy composition and execution of bioinformatics workflows.

Chapter 24 focuses on the design and implementation of a critical computer program in structural biology onto two computational and data grids. The first is the Buffalo-based ACDC grid, which uses facilities at SUNY–Buffalo and several research institutions in the greater Buffalo area. The second is Grid2003, an international grid established late in 2003 primarily for physics and astronomy applications. The authors present an overview of the ACDC Grid and Grid2003, focusing on the implementation of several new tools that they have developed for the integration of computational and data grids, lightweight job monitoring, predictive scheduling, and opportunities for improved grid utilization through an elegant backfill facility. A new computational framework is developed for the evolutionary determination, an efficient implementation of an algorithm to determine molecular crystal structures using the Shake-and-Bake methodology. Finally, the grid-enabled data mining approach that the authors introduce is able to exploit computational cycles that would otherwise go unused.

Recently, there has been an increase in the number of completely sequenced genomes due to the numerous genome-sequencing projects. The enormous biological sequence data thus generated necessitate the development of efficient tools for mining the information on structural and functional properties of biomolecules. Such a kind of information can prove invaluable for pharmaceutical industries, for in silico drug target identification and new drug discovery. However, the enormity of data and complexity of algorithms make the above tasks computationally demanding, necessitating the use of high-performance computing. Lately, the cost-effective general-purpose clusters of PCs and workstations have been gaining importance in bioinformatics. However, to use these techniques one must still have significant expertise not only in the bioinformatics domain but also in parallel computing. A problem-solving environment (PSE) relieves the scientist of the burdens associated with the needless and often confidential details of the hardware and software systems by providing a user-friendly environment either through web portals or graphical user interfaces. The PSE thus leaves the scientist free to concentrate on the job. This chapter describes the design and development of GIPSY, a PSE for bioinformatics applications.

Chapter 26 describes the TaskSpaces software framework for grid computing. TaskSpaces is characterized by two major design choices: decentralization, provided by an underlying tuple space concept, and platform independence, provided by implementation in Java. This chapter discusses advantages and disadvantages of this approach, and demonstrate seamless performance on an ad hoc grid composed of a wide variety of hardware for a real-life parallel bioinformatics problem. Specifically, the authors performed virtual experiments in RNA folding on computational grids composed of fast supercomputers, to estimate the smallest pool of random RNA molecules that would contain enough catalytic motifs for starting a primitive metabolism. These experiments may establish one of the missing links in the chain of events that led to the origin of life.

Desktop grids have been used to perform some of the largest computations in the world and have the potential to grow by several orders of magnitude. However, current approaches to using desktop resources require either centralized servers or extensive knowledge of the underlying system, limiting their scalability. The authors propose a new design for desktop grids that relies on a self-organizing, fully decentralized approach to the organization of the computation. Their approach, called the Organic Grid, is a radical departure from current approaches and is modeled after the way complex biological systems organize themselves. Similar to current desktop grids, a large computational task is broken down into sufficiently small subtasks. Each subtask is encapsulated into a mobile agent, which is then released on the grid and discovers computational resources using autonomous behavior. In the process of "colonization" of available resources, the judicious design of the agent behavior produces the emergence of crucial properties of the computation that can be tailored to specific classes of applications. The authors demonstrate this concept with a reduced-scale proof-of-concept implementation that executes a data-intensive independent-task application on a set of heterogeneous, geographically distributed machines. They present a detailed exploration of the design space of our system and a performance evaluation of our implementation using metrics appropriate for assessing self-organizing desktop grids.

A new computing approach is introduced in Chapter 28 that makes use of field programmable gate arrays (FPGAs). This new approach uses FPGA processors that are integrated into existing computing nodes. The FPGA processors provide a computing structure that enables to execute the algorithms in a parallel architecture.

The transformation from the sequential algorithm to the parallel architecture is described by the energy calculation part of a protein structure prediction task.

Technological advances in microscopy, digital image acquisition, and automation have allowed digital, virtual slides to be used in pathology and microbiology. Virtual microscopy has the benefits of parallel distribution, on-demand reviews, rapid diagnosis, and long-term warehousing of slides. Sensor technologies combined with high-power magnification generate uncompressed images that can reach 50 GB per image in size. In a clinical or research environment, the number of slides scanned can compound the challenges in storing and managing these images. A distributed storage system coupled with a distributed execution framework is currently the best way to overcome these challenges to perform large-scale analysis and visualization. Chapter 29 demonstrates an implementation that integrates several middleware components in a distributed environment to enable and optimize the storage and analysis of this digital information. These systems support and enable virtual slide reviews, pathology image analysis, and three-dimensional reconstruction and visualization of microscopy data sets in both clinical and research settings.

ALBERT Y. ZOMAYA

CONTRIBUTORS

David Abramson, Monash University, Clayton, Victoria, Australia

Enrique Alba, Universidad de Málaga, Málaga, Spain

Ali Al Mazari, The University of Sydney, Sydney, Australia

Ilkay Altintas, University of California, San Diego, California, USA

Celine Amoreira, University of Zurich, Zurich, Switzerland

R. Andonov, Campus de Beaulieu, Rennes, France

Santosh Atanur, Pune University, Pune, India

David A. Bader, University of New Mexico, Albuquerque, New Mexico, USA

Kim K. Baldridge, University of Zurich, Zurich, Switzerland and University of California, San Diego, California, USA

S. Balev, Université du Havre, Le Havre, France

Gerald Baumgartner, The Ohio State University, Columbus, Ohio, USA

Dattatraya Bhat, Pune University, Pune, India

Adam Birnbaum, University of California, San Diego, California, USA

Shahid H. Bokhari, University of Engineering and Technology, Lahore, Pakistan

Azzedine Boukerche, University of Ottawa, Ottawa, Ontario, Canada

K. Burrage, University of Queensland, Queensland, Australia

P. M. Burrage, University of Queensland, Queensland, Australia

Eric S. Carlson, University of Alabama, Auburn, Alabama, USA

U. Catalyurek, The Ohio State University, Columbus, Ohio, USA

Arjav J. Chakravarti, The MathWorks, Natick, Massachusetts, USA

Christophe Chassagnole, Institut National de Sciences Appliquées, Lyon, France

Vipin Chaudhary, Wayne State University, Troy, Michigan, USA

Janaki Chintalapati, Pune University, Pune, India

D. Cowden, The Ohio State University, Columbus, Ohio, USA

Jack da Silva, The University of Adelaide, Adelaide, Australia

Amitava Datta, University of Western Australia, Perth, Australia

Richard O. Day, Air Force Institute of Technology, Wright-Patterson Air Force Base, Dayton, Ohio, USA

Alba Cristina Magalhaes Alves de Melo, Universidade de Brasilia, Brasil

Hans De Sterck, University of Waterloo, Waterloo, Ontario, Canada

Clarisse Dhaenens, Universite des Sciences et Technologies de Lille, Lille, France

Andrei Doncescu, Laboratory of Analysis and Architecture of Systems LAAS CNRS 8001, Toulouse, France

Justin Ebedes, University of Western Australia, Perth, Australia

Colin Enticott, Monash University, Clayton, Victoria, Australia

Slavisa Garic, Monash University, Clayton, Victoria, Australia

Mark L. Green, State University of New York, Buffalo, New York, USA

Jerry P. Greenberg, University of California, San Diego, California, USA

N. Hamilton, University of Queensland, Queensland, Australia

S. Hastings, The Ohio State University, Columbus, Ohio, USA

Sameer Ingle, Pune University, Pune, India

S. Jewel, The Ohio State University, Columbus, Ohio, USA

Calvin A. Johnson, National Institutes of Health, Bethesda, Maryland, USA

Rajendra R. Joshi, Centre for Development of Advanced Computing, Ganeshkhind, Maharashtra, India

Ning Kang, University of Kentucky, Lexington, Kentucky, USA

Mohammed Khabzaoui, Universite des Sciences et Technologies de Lille, Lille, France

Sami Khuri, San Jose State University, San Jose, California, USA

Rob Knight, University of Colorado at Boulder, Boulder, Colorado, USA

Arun Krishnan, Bioinformatics Institute, Matrix, Singapore

T. Kurc, The Ohio State University, Columbus, Ohio, USA

Gary B. Lamont, Air Force Institute of Technology, Wright-Patterson Air Force Base, Dayton, Ohio, USA

S. Langella, The Ohio State University, Columbus, Ohio, USA

Mario Lauria, The Ohio State University, Columbus, Ohio, USA

Feng Liu, Wayne State University, Troy, Michigan, USA

Gabriel Luque, Universidad de Málaga, Málaga, Spain

Rob Markel, National Center for Atmospheric Research, Boulder, Colorado, USA

Robert L. Martino, National Institutes of Health, Laurel, Maryland, USA

Vijay Matta, Wayne State University, Troy, Michigan, USA

Xiandong Meng, Wayne State University, Troy, Michigan, USA

Daniel Merkle, Universität Leipzig, Leipzig, Germany

Martin Middendorf, Universität Leipzig, Leipzig, Germany

Russ Miller, State University of New York, Buffalo, New York, USA

Bernard M.E. Moret, University of New Mexico, Albuquerque, New Mexico, USA

Satish Mummadi, Pune University, Pune, India

Anil Nambiar, Wayne State University, Troy, Michigan, USA

S. Oster, The Ohio State University, Columbus, Ohio, USA

T. Pan, The Ohio State University, Columbus, Ohio, USA

Ekkehard Petzold, MPI fur Evolutionare Anthropologie, Germany

Yohann Potier, University of Zurich, Zurich, Switzerland

Jithesh P.V., Pune University, Pune, India

Nouhad J. Rizk, Notre Dame University, Zouk Mosbeh, Lebanon

Juan Carlos A. Rodríguez, University of Barcelona, Barcelona, Spain

Daniel E. Russ, National Institutes of Health, Bethesda, Maryland, USA

J. Saltz, The Ohio State University, Columbus, Ohio, USA

Jon R. Sauer, Eagle Research & Development, Boulder, Colorado, USA

Bertil Schmidt, Nanyang Technological University, Singapore

Heiko A. Schmidt, Institut fuer Bioinformatik, Duesseldorf, Germany

Heiko Schröder, RMIT University, Melbourne, Australia

Harald Simmler, Bgm.-Horlacherstr., Ludwigshafen, Germany

Uddhavesh Sonavane, Pune University, Pune, India

Alexandros Stamatakis, Institut fur Informatik, Technische Universitat, Munchen, Germany

Wibke Sudholt, University of Zurich, Switzerland and Computational Laboratory, ETH Zurich, Switzerland

El-Ghazali Talbi, LIFL — University of Lille, Villeneuve d'Ascq, France

T. Tian, University of Queensland, Queensland, Australia

Arndt von Haeseler, Bioinformatik, HHU Dusseldorf and von-Neumann Institut fur Computing, NA, Germany

Chau-Wen Tseng, University of Maryland at College Park, Maryland, USA

Tiffani L. Williams, Radcliffe Institute, Cambridge, Massachusetts, USA

Xue Wu, University of Maryland at College Park, Maryland, USA

Ganesh Yadav, Wayne State University, Troy, Michigan, USA

Mi Yan, Texas A&M University, College Station, Texas, USA

N. Yanev, University of Sofia, Bulgaria

Laurence T. Yang, St. Francis Xavier University, Antigonish, Nova Scotia, Canada

Jun Zhang, University of Kentucky, Lexington, Kentucky, USA

Ruhong Zhou, IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA

Albert Y. Zomaya, Sydney University, Sydney, NSW, Australia

ACKNOWLEDGMENTS

First and foremost I would like to thank and acknowledge the contributors to this volume for their support and patience, and the reviewers for their useful comments and suggestions that helped in improving the earlier outline of the book and presentation of the material. Also, I should extend my deepest thanks to Val Moliere and Emily Simmons from Wiley for their collaboration, guidance, and most importantly, patience in finalizing this book. Finally, I would like to acknowledge the efforts of the team from Wiley's production department for their extensive efforts during the many phases of this project and the timely fashion in which the book was produced by.

PART I

ALGORITHMS AND MODELS

CHAPTER 1

Parallel and Evolutionary Approaches to Computational Biology

NOUHAD J. RIZK

Many of the today's problems, such as those involved in weather prediction, aerodynamics, and genetic mapping, require tremendous computational resources to be solved accurately. These applications are computationally very intensive and require vast amounts of processing power and memory requirements. Therefore, to give accurate results, powerful computers are needed to reduce the run time, for example, finding genes in DNA sequences, predicting the structure and functions of new proteins, clustering proteins into families, aligning similar proteins, and generating phylogenetic trees to examine evolutionary relationships all need complex computations. To develop parallel computing programs for such kinds of computational biology problems, the role of a computer architect is important; his or her role is to design and engineer the various levels of a computer system to maximize performance and programmability within limits of technology and cost. Thus, parallel computing is an effective way to tackle problems in biology; multiple processors being used to solve the same problem. The scaling of memory with processors enables the solution of larger problems than would be otherwise possible, while modeling a solution is as much important as the computation.

In this chapter, after an introduction to genes and genomes, we describe some efficient parallel algorithms that efficiently solve applications in computational biology. An evolutionary approach to computational biology is presented based first on the search space, which is the set of all possible solutions. The second factor used for the formulation of an optimization problem is the determination of a fitness function that measures how good a particular answer is. Finally, a significant deviation from the standard parallel solution to genetic parallel algorithms approach theory is pointed out by arguing that parallel computational biology is an important sub-discipline that merits significant research attention and that combining different solution paradigms is worth implementing.

Parallel Computing for Bioinformatics and Computational Biology, Edited by Albert Y. Zomaya

1.1 INTRODUCTION

Computational biology is the use of computational techniques to model biological systems at various levels of complexity — atomic, metabolic, cellular, and pathologic. The field of computational biology covers many areas: structural biology, biochemistry, physical chemistry, molecular biology, genomics and bioinformatics, control theory, statistics, mathematics, and computer science. Bioinformatics provides a wealth of potential challenges that can be used to advance the state of the art by creating scalable applications that can be used in customer environments. Thus, in computational biology, conducting research related to the realization of parallel/distributed scalable applications requires an understanding of the basics of all related fields. Therefore, this chapter starts with a detailed explanation of certain technical terms that have proved to be essential for researchers in computational biology.

1.1.1 Chromosome

A chromosome is a long string of double-stranded deoxyribonucleic acid (DNA), the molecule that serves as a primary repository of genetic information. Thomas Hunt Morgan found that genes on a chromosome have a remarkable statistical property, that is, genes appear as being linearly arranged along the chromosome and also that chromosomes can recombine and exchange genetic material. A gene is a unit of heredity used to describe a unit of phenotype variation.

1.1.2 Allele

Alleles are alternate forms of the same gene. There may be hundreds of alleles for a particular gene, but usually only one or a few are common. A homologous pair of chromosomes contain two alleles, one in the chromosome derived from the father and the other in the chromosome derived from the mother. If, for example, the chromosome inherited from the mother has a mutant allele at a specific position, this position on a chromosome is called a locus, and the presence of a single mutant allele creates the trait of disease. However, the child will not suffer from the disease caused by this mutation unless both the genes inherited from parents are defective or one of them is on the X chromosome, for example, hemophilus. In brief, an allele is a type of the DNA at a particular locus on a particular chromosome.

1.1.3 Recombination

Recombination or crossing over is defined as the recombination of maternal chromosome pairs with its paternal chromosome and exchanges material in the genesis of a sperm or egg. This formation of new gene combination is the result of the physical event of crossing over. The intensity of linkage of two genes can be measured by the frequency of the recombinants. The probability that a recombination event occurs between two loci is a function of the distance between these loci. In fact, the alleles at two loci that are far apart on a chromosome are more likely to combine than the alleles that are close together on a chromosome. Genes that tend to stay together during recombination are called linked. Sometimes, one gene in a linked pair serves as a *marker* that can be used by geneticists to infer the presence of the other genes causing disease.

1.1.4 Meiosis

Before explaining meiosis, let us explain the relationship between genes and alleles. In Figure 1.1, we notice two gametes inherited from the father AD, which are called the gene 1, and the two gametes inherited from the mother ad, which are called gene 2. Therefore, the formation of haploid germ cells from diploid parent cell is called meiosis. Meiosis is informative for linkage when we identify whether the gamete is recombinant.

1.1.5 Genetic Linkage

Geneticists seek to locate genes for disorder traits (gene disease) among the genome, which is pairs of 23 human chromosomes. The statistical procedure used to trace the transmission of a disordered allele within a family is called linkage analysis. This analysis is based on genes, whose locations on a particular chromosome are already known, and are called markers [1].

Genes will be inherited together if they are close on the same chromosome because recombination is less likely. Recombinant chromosomes will occur less frequently

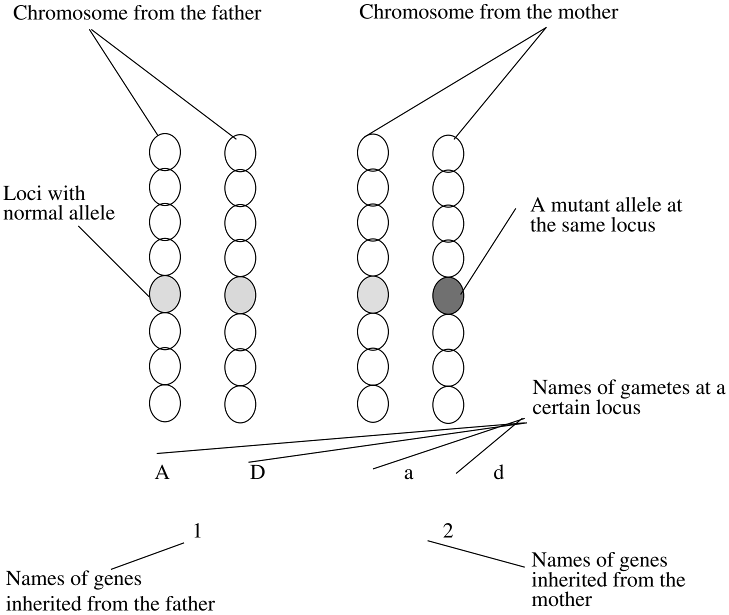
Chromosome from the father
Chromosome from the mother
Loci with normal allele
A mutant allele at the same locus
Names of gametes at a certain locus
A
D
a
d
1
2
Names of genes inherited from the father
Names of genes inherited from the mother

Figure 1.1 Genes and alleles at a specific locus.

TABLE 1.1 Expected Frequency of Children Through the Mother

Mother's Gametes	Recombinant	Probability
Ad	No	1
Ad	No	1
Ad	No	1
Ad	No	1

(less than half of the time) than nonrecombinant chromosomes (more than half of the time). The recombination is measured by the recombination fraction denoted by θ, which is the probability of a recombination between two loci. The lowest possible value for θ is zero, which means that there is no recombination between the marker and the trait locus. In fact, either the marker is the trait or the two loci are so close together that they rarely recombine. The upper limit of θ is 0.5, which means that the loci are not linked. The recombination frequency θ for unlinked loci is 0.5. In brief, $0 \leq \theta \leq 0.5$ [2].

The family is said to be idealized pedigree if no recombination has taken place. If there are recombinations, it is quite easy to calculate θ; it is the summation of the number of recombinants among offspring divided by the total number of offspring. For example, if one recombinant out of nine offspring means that θ is equal to $1/9 = 0.11$.

1.1.6 Expected Frequency of Offspring

The computation of expected frequency of offspring genotypes in linkage is as follows [3]: As an example, consider a father who has haplotype AD/ad and a mother with haplotype ad/ad. All the mother's gametes will be genotyped ad. Thus, the probability that the mother gives the alleles ad is equal to 1 (see Table 1.1).

Father, however, may have one of the four different gametes, AD, Ad, aD, ad. In addition, the probability that the father gives the alleles AD is equal to the probability that the father gives allele A at marker multiplied by the probability of having no recombination between marker and trait. In fact, it is equal to $1/2(1 - \theta)$. Similarly, the probability that the father gives the alleles Ad is equal to the probability that the father gives allele A at marker multiplied by the probability of having recombination between marker and trait. In fact, it is equal to $1/2\theta$. The probability that the father

TABLE 1.2 Expected Frequency of Children Through the Father

Father's Gametes	Recombinant	Probability
AD	No	$1/2(1 - \theta)$
Ad	Yes	$1/2\theta$
AD	Yes	$1/2\theta$
Ad	No	$1/2(1 - \theta)$

TABLE 1.3 **Expected Frequency of Children**

Father's and Mother's Gametes	Recombinant	Probability
AD/ad	No	$1/2(1-\theta)$
AD/ad	Yes	$1/2\theta$
aD/ad	Yes	$1/2\theta$
ad/ad	No	$1/2(1-\theta)$

gives the alleles aD is equal to the probability that the father gives allele a at marker multiplied by the probability of having no recombination between marker and trait. In fact, it is equal to $1/2\theta$. Finally, the probability that the father gives the alleles ad is equal to the probability that the father gives allele a at marker multiplied by the probability of having recombination between marker and trait. In fact, it is equal to $1/2(1-\theta)$ (see Tables 1.2–1.4). Then, the expected frequency among the offspring is a function of θ.

1.1.7 Multipoint Linkage Analysis

In the previous section, we assumed that we know where is the gene affected but what if we do not know? Therefore, we need to gather a large number of families in which we observe a disorder and we extract some biological specimen from each member of the family to study the linkage, but this time with many markers simultaneously; this procedure is called multipoint linkage [4]. There are two types of statistical techniques used in the linkage analysis, parametric linkage analysis and nonparametric linkage analysis. Parametric linkage analysis uses statistical procedures to estimate θ and sometimes other quantities. The odds for linkage is a quantity that is equal to the ratio of two probabilities; the numerator is the probability of observing the data given that θ is less than 0.5 (i.e., the marker and the trait loci are linked) and the denominator is the probability of observing the data given that θ is equal to 0.5 (i.e., the marker and the trait loci are not linked). The common logarithm (base 10) of the odds (likelihood) of linkage is specific to geneticists for the computation of parametric linkage analysis. It is called the LOD scores. The second method used in linkage analysis is suitable for complex gene disorders, unlike the first one suitable for single gene analysis. It is called the nonparametric approach. The advantages of nonparametric techniques are that it is not necessary to make assumptions about the mode of inheritance for the disorder; their disadvantage is they are less powerful than

TABLE 1.4 **Expected Frequency Function of Theta**

Offspring	Recombinant	Probability	$\theta = 0$	$\theta = 0.10$	$\theta = 0.2$	$\theta = 0.3$	$\theta = 0.4$	$\theta = 0.5$
AD	No	$1/2(1-\theta)$	0.5	0.45	0.40	0.35	0.30	0.25
Ad	Yes	$1/2\theta$	0.00	0.05	0.10	0.15	0.20	0.25
aD	Yes	$1/2\theta$	0.00	0.05	0.10	0.15	0.20	0.25
ad	No	$1/2(1-\theta)$	0.50	0.45	0.40	0.35	0.30	0.25

the parametric techniques. An example of nonparametric techniques is the affected sib-pair method. The geneticists gather data on a large number of sibships to locate those that have at least two members of a sibship who are affected with the disorder. The affected sib pairs are then genotyped at the marker locus, and the sib pairs are placed into one of the two mutually exclusive categories based on their genotypes at the marker. The first category includes all sib pairs who have the same genotype at the marker, these being called marker-concordant pairs. The second category is for the marker-discordant pairs, those sib pairs who have different genotypes at the marker. If the marker is not linked to the gene for the disorder, then we should expect an equal number in both categories. However, if the marker is linked to the disease locus, then there should be more marker-concordant pairs than marker-discordant pairs.

Sib Pair Analysis The sib pair analysis is the probability of having 0, 1 or 2 common alleles. This analysis is known as identity by descent (IBD) (Fig. 1.2). Consider a sib pair and suppose we wish to identify the parental origin of the DNA inherited by each sib at a particular locus, say. Label the paternal chromosomes containing the locus of interest by (a, c), and similarly label the maternal chromosomes by (b, d).

The inheritance vector of the sib-pair at the locus l is the vector $x = (x_1, x_2, x_3, x_4)$, where x

- x_1 is the label of the paternal chromosome from which sib1 inherited DNA at locus l (a),
- x_2 is the label of the maternal chromosome from which sib1 inherited DNA at locus l (b),
- x_3 is the label of the paternal chromosome from which sib2 inherited DNA at locus l (c), and
- x_4 is the label of the maternal chromosome from which sib1 inherited DNA at locus l (d).

In practice, the inheritance vector of a sibship is determined by finding enough polymorphism in the parents to be able to identify the chromosomal fragments transmitted

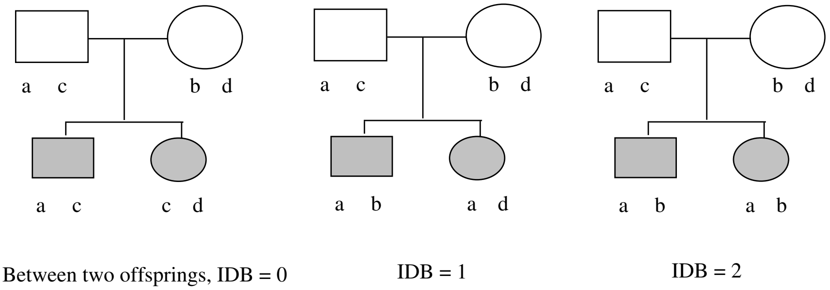

Figure 1.2 Identity by descent (IBD).