PARALLEL COMPUTING
FOR BIOINFORMATICS
AND COMPUTATIONAL
BIOLOGY
MODELS, ENABLING TECHNOLOGIES,
AND CASE STUDIES

Edited by
Albert Y. Zomaya
The University of Sydney, Australia
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
To our families for their help, support, and patience.

Albert Zomaya
## CONTENTS

Preface xv  
Contributors xxv  
Acknowledgments xxix

### PART I ALGORITHMS AND MODELS 1

1 **Parallel and Evolutionary Approaches to Computational Biology** 3  
   *Nouhad J. Rizk*  
   1.1 Introduction 4  
   1.2 Bioinformatics 13  
   1.3 Evolutionary Computation Applied to Computational Biology 20  
   1.4 Conclusions 23  
   References 25

2 **Parallel Monte Carlo Simulation of HIV Molecular Evolution in Response to Immune Surveillance** 29  
   *Jack da Silva*  
   2.1 Introduction 30  
   2.2 The Problem 30  
   2.3 The Model 32  
   2.4 Parallelization with MPI 39  
   2.5 Parallel Random Number Generation 43  
   2.6 Preliminary Simulation Results 46  
   2.7 Future Directions 52  
   References 55

3 **Differential Evolutionary Algorithms for In Vivo Dynamic Analysis of Glycolysis and Pentose Phosphate Pathway in *Escherichia coli*** 59  
   *Christophe Chassagnole*  
   3.1 Introduction 59  
   3.2 Mathematical Model 61  
   3.3 Estimation of the Parameters of the Model 67  
   3.4 Kinetic Parameter Estimation by DE 69
## CONTENTS

3.5 Simulation and Results 70
3.6 Stability Analysis 73
3.7 Control Characteristic 73
3.8 Conclusions 75
References 76

4 **Compute-Intensive Simulations for Cellular Models** 79  
*K. Burrage*

4.1 Introduction 79
4.2 Simulation Methods for Stochastic Chemical Kinetics 81
4.3 Aspects of Biology — Genetic Regulation 92
4.4 Parallel Computing for Biological Systems 96
4.5 Parallel Simulations 100
4.6 Spatial Modeling of Cellular Systems 104
4.7 Modeling Colonies of Cells 109
References 115

5 **Parallel Computation in Simulating Diffusion and Deformation in Human Brain** 121  
*Ning Kang*

5.1 Introduction 121
5.2 Anisotropic Diffusion Simulation in White Matter Tractography 122
5.3 Brain Deformation Simulation in Image-Guided Neurosurgery 132
5.4 Summary 142
References 143

### PART II SEQUENCE ANALYSIS AND MICROARRAYS 147

6 **Computational Molecular Biology** 149  
*Azzedine Boukerche*

6.1 Introduction 149
6.2 Basic Concepts in Molecular Biology 150
6.3 Global and Local Biological Sequence Alignment 152
6.4 Heuristic Approaches for Biological Sequence Comparison 158
6.5 Parallel and Distributed Sequence Comparison 161
6.6 Conclusions 164
References 165

7 **Special-Purpose Computing for Biological Sequence Analysis** 167  
*Bertil Schmidt*

7.1 Introduction 167
7.2 Hybrid Parallel Computer 169
CONTENTS

7.3 Dynamic Programming Communication Pattern 172
7.4 Performance Evaluation 179
7.5 Future Work and Open Problems 185
7.6 Tutorial 188
References 190

8 Multiple Sequence Alignment in Parallel on a Cluster of Workstations 193
Amitava Datta

8.1 Introduction 193
8.2 CLUSTAL W 194
8.3 Implementation 201
8.4 Results 207
8.5 Conclusion 209
References 210

9 Searching Sequence Databases Using High-Performance BLASTs 211
Xue Wu

9.1 Introduction 211
9.2 Basic Blast Algorithm 212
9.3 Blast Usage and Performance Factors 214
9.4 High Performance BLASTs 215
9.5 Comparing BLAST Performance 221
9.6 UMD-BLAST 226
9.7 Future Directions 228
9.8 Related Work 229
9.9 Summary 230
References 230

10 Parallel Implementations of Local Sequence Alignment: Hardware and Software 233
Vipin Chaudhary

10.1 Introduction 233
10.2 Sequence Alignment Primer 235
10.3 Smith–Waterman Algorithm 240
10.4 FASTA 244
10.5 BLAST 245
10.6 HMMER — Hidden Markov Models 249
10.7 ClustalW 252
10.8 Specialized Hardware: FPGA 257
10.9 Conclusion 262
References 262
CONTENTS

11 Parallel Computing in the Analysis of Gene Expression Relationships 265
Robert L. Martino

11.1 Significance of Gene Expression Analysis 265
11.2 Multivariate Gene Expression Relations 267
11.3 Classification Based on Gene Expression 274
11.4 Discussion and Future Directions 280
References 282

12 Assembling DNA Fragments with a Distributed Genetic Algorithm 285
Gabriel Luque

12.1 Introduction 285
12.2 DNA Fragment Assembly Problem 286
12.3 DNA Fragment Assembly Using the Sequential GA 289
12.4 DNA Fragment Assembly Problem Using the Parallel GA 292
12.5 Experimental Results 294
12.6 Conclusions 301
References 301

13 A Cooperative Genetic Algorithm for Knowledge Discovery in Microarray Experiments 303
Mohammed Khabzaoui

13.1 Introduction 303
13.2 Microarray Experiments 304
13.3 Association Rules 306
13.4 Multi-Objective Genetic Algorithm 308
13.5 Cooperative Multi-Objective Genetic Algorithm (PMGA) 313
13.6 Experiments 315
13.7 Conclusion 322
References 322

PART III PHYLOGENETICS 325

14 Parallel and Distributed Computation of Large Phylogenetic Trees 327
Alexandros Stamatakis

14.1 Introduction 327
14.2 Maximum Likelihood 330
14.3 State-of-the-Art ML Programs 332
14.4 Algorithmic Solutions in RAxML-III 334
14.5 HPC Solutions in RAxML-III 337
14.6 Future Developments 341
References 344
15 Phylogenetic Parameter Estimation on COWs
Ekkehard Petzold

15.1 Introduction 347
15.2 Phylogenetic Tree Reconstruction using Quartet Puzzling 349
15.3 Hardware, Data, and Scheduling Algorithms 354
15.4 Parallelizing PEst 356
15.5 Extending Parallel Coverage in PEst 359
15.6 Discussion 365
References 367

16 High-Performance Phylogeny Reconstruction Under Maximum Parsimony
Tiffani L. Williams

16.1 Introduction 369
16.2 Maximum Parsimony 374
16.3 Exact MP: Parallel Branch and Bound 378
16.4 MP Heuristics: Disk-Covering Methods 381
16.5 Summary and Open Problems 390
References 392

PART IV PROTEIN FOLDING

17 Protein Folding with the Parallel Replica Exchange Molecular Dynamics Method
Ruhong Zhou

17.1 Introduction 397
17.2 REMD Method 399
17.3 Protein Folding with REMD 403
17.4 Protein Structure Refinement with REMD 420
17.5 Summary 422
References 423

18 High-Performance Alignment Methods for Protein Threading
R. Andonov

18.1 Introduction 427
18.2 Formal Definition 431
18.3 Mixed Integer Programming Models 434
18.4 Divide-and-Conquer Technique 444
18.5 Parallelization 448
18.6 Future Research Directions 453
18.7 Conclusion 454
18.8 Summary 454
References 455
CONTENTS

19 Parallel Evolutionary Computations in Discerning Protein Structures 459

Richard O. Day

19.1 Introduction 459
19.2 PSP Problem 460
19.3 Protein Structure Discerning Methods 461
19.4 PSP Energy Minimization EAs 471
19.5 PSP Parallel EA Performance Evaluation 477
19.6 Results and Discussion 479
19.7 Conclusions and Suggested Research 483

References 483

PART V PLATFORMS AND ENABLING TECHNOLOGIES 487

20 A Brief Overview of Grid Activities for Bioinformatics and Health Applications 489

Ali Al Mazari

20.1 Introduction 489
20.2 Grid Computing 490
20.3 Bioinformatics and Health Applications 491
20.4 Grid Computing for Bioinformatics and Health Applications 491
20.5 Grid Activities in Europe 492
20.6 Grid Activities in the United Kingdom 494
20.7 Grid Activities in the USA 497
20.8 Grid Activities in Asia and Japan 498
20.9 International Grid Collaborations 499
20.10 International Grid Collaborations 499
20.11 Conclusions and Future Trends 500

References 501

21 Parallel Algorithms for Bioinformatics 509

Shahid H. Bokhari

21.1 Introduction 509
21.2 Parallel Computer Architecture 511
21.3 Bioinformatics Algorithms on the Cray MTA System 517
21.4 Summary 527

References 528

22 Cluster and Grid Infrastructure for Computational Chemistry and Biochemistry 531

Kim K. Baldridge

22.1 Introduction 531
22.2 GAMESS Execution on Clusters 532
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>22.3</td>
<td>Portal Technology</td>
<td>537</td>
</tr>
<tr>
<td>22.4</td>
<td>Running GAMESS with Nimrod Grid-Enabling Infrastructure</td>
<td>538</td>
</tr>
<tr>
<td>22.5</td>
<td>Computational Chemistry Workflow Environments</td>
<td>542</td>
</tr>
<tr>
<td>22.6</td>
<td>Conclusions</td>
<td>546</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>548</td>
</tr>
<tr>
<td>23</td>
<td>Distributed Workflows in Bioinformatics</td>
<td>551</td>
</tr>
<tr>
<td></td>
<td><em>Arun Krishnan</em></td>
<td></td>
</tr>
<tr>
<td>23.1</td>
<td>Introduction</td>
<td>551</td>
</tr>
<tr>
<td>23.2</td>
<td>Challenges of Grid Computing</td>
<td>553</td>
</tr>
<tr>
<td>23.3</td>
<td>Grid Applications</td>
<td>554</td>
</tr>
<tr>
<td>23.4</td>
<td>Grid Programming</td>
<td>555</td>
</tr>
<tr>
<td>23.5</td>
<td>Grid Execution Language</td>
<td>557</td>
</tr>
<tr>
<td>23.6</td>
<td>GUI-Based Workflow Construction and Execution</td>
<td>565</td>
</tr>
<tr>
<td>23.7</td>
<td>Case Studies</td>
<td>570</td>
</tr>
<tr>
<td>23.8</td>
<td>Summary</td>
<td>578</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>579</td>
</tr>
<tr>
<td>24</td>
<td>Molecular Structure Determination on a Computational and Data Grid</td>
<td>583</td>
</tr>
<tr>
<td></td>
<td><em>Russ Miller</em></td>
<td></td>
</tr>
<tr>
<td>24.1</td>
<td>Introduction</td>
<td>583</td>
</tr>
<tr>
<td>24.2</td>
<td>Molecular Structure Determination</td>
<td>585</td>
</tr>
<tr>
<td>24.3</td>
<td>Grid Computing in Buffalo</td>
<td>586</td>
</tr>
<tr>
<td>24.4</td>
<td>Center for Computational Research</td>
<td>588</td>
</tr>
<tr>
<td>24.5</td>
<td>ACDC-Grid Overview</td>
<td>588</td>
</tr>
<tr>
<td>24.6</td>
<td>Grid Research Collaborations</td>
<td>596</td>
</tr>
<tr>
<td>24.7</td>
<td>Grid Research Advancements</td>
<td>601</td>
</tr>
<tr>
<td>24.8</td>
<td>Grid Research Application Abstractions and Tools</td>
<td>603</td>
</tr>
<tr>
<td>24.9</td>
<td>Conclusions</td>
<td>616</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>616</td>
</tr>
<tr>
<td>25</td>
<td>GIPSY: A Problem-Solving Environment for Bioinformatics Applications</td>
<td>623</td>
</tr>
<tr>
<td></td>
<td><em>Rajendra R. Joshi</em></td>
<td></td>
</tr>
<tr>
<td>25.1</td>
<td>Introduction</td>
<td>623</td>
</tr>
<tr>
<td>25.2</td>
<td>Architecture</td>
<td>626</td>
</tr>
<tr>
<td>25.3</td>
<td>Currently Deployed Applications</td>
<td>634</td>
</tr>
<tr>
<td>25.4</td>
<td>Conclusion</td>
<td>647</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>648</td>
</tr>
<tr>
<td>26</td>
<td>TaskSpaces: A Software Framework for Parallel Bioinformatics on</td>
<td>651</td>
</tr>
<tr>
<td></td>
<td>Computational Grids</td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>Hans De Sterck</em></td>
<td></td>
</tr>
<tr>
<td>26.1</td>
<td>Introduction</td>
<td>651</td>
</tr>
<tr>
<td>26.2</td>
<td>The TaskSpaces Framework</td>
<td>655</td>
</tr>
</tbody>
</table>
CONTENTS

26.3 Application: Finding Correctly Folded RNA Motifs 661
26.4 Case Study: Operating the Framework on a Computational Grid 663
26.5 Results for the RNA Motif Problem 664
26.6 Future Work 668
26.7 Summary and Conclusion 669
References 669

27 The Organic Grid: Self-Organizing Computational Biology on Desktop Grids 671
Arjav J. Chakravarti

27.1 Introduction 672
27.2 Background and Related Work 674
27.3 Measurements 686
27.4 Conclusions 698
27.5 Future Directions 699
References 700

28 FPGA Computing in Modern Bioinformatics 705
H. Simmler

28.1 Parallel Processing Models 706
28.2 Image Processing Task 708
28.3 FPGA Hardware Accelerators 711
28.4 Image Processing Example 716
28.5 Case Study: Protein Structure Prediction 720
28.6 Conclusion 733
References 734

29 Virtual Microscopy: Distributed Image Storage, Retrieval, Analysis, and Visualization 737
T. Pan

29.1 Introduction 737
29.2 Architecture 738
29.3 Image Analysis 747
29.4 Clinical Use 752
29.5 Education 755
29.6 Future Directions 756
29.7 Summary 759
References 760

Index 765
Bioinformatics and Computational Biology are fields that require 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 overtake 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 Saitou 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 $\sigma$-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 phylogenetics) 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
CONTRIBUTORS

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. Ku rc, 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 für Evolutionäre 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 für Bioinformatik, Duesseldorf, Germany
Heiko Schröder, RMIT University, Melbourne, Australia
Harald Simmler, Bgm.-Horlacherstr., Ludwigshafen, Germany
Uddhavesh Sonavane, Pune University, Pune, India
CONTRIBUTORS

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