Preface

Some of the most challenging and influential opportunities for computational scientists arise in developing and applying information technology to understand the molecular machinery of the cell. The past decade has shown that many novel algorithms may be developed and fruitfully applied to the challenges of computational molecular biology. This research has led to computer systems and algorithms that are useful in structural molecular biology, proteomics, and rational drug design.

Concomitantly, a wealth of interesting computational problems arise in proposed methods for discovering new pharmaceuticals. These include algorithms for interpreting X-ray crystallography and nuclear magnetic resonance (NMR) data, disease classification using mass spectrometry of human serum, and protein design. For example, this book presents chapters on provable algorithms that have recently been used to reveal the enzymatic architecture of organisms high on the CDC bioterrorism watch list (chapter 47), for probabilistic cancer classification from human peripheral blood (chapter 19), and to redesign an antibiotic-producing enzyme to adenylate a novel substrate (chapter 12).

In the postgenomic era, key problems in molecular biology center on the determination and exploitation of three-dimensional protein structure and function. For example, modern drug design techniques use protein structure to understand how a drug can bind to an enzyme and inhibit its function. Structural proteomics will require high-throughput experimental techniques, coupled with sophisticated computer algorithms for data analysis and experiment planning.

Novel computational methods are enabling high-throughput structural and functional studies of proteins. One recent subfield is structural genomics, whose goal is (in the broadest terms) to determine the three-dimensional structures of all proteins in nature, through a combination of direct experiments and theoretical analysis. By determining the structures of proteins, we are better able to understand how each protein functions normally and how faulty protein structures can cause disease. Scientists can use the structures of disease-related proteins to help develop new therapeutics and diagnostic techniques.

At the molecular level, many genes provide the blueprint for proteins, and it remains very expensive and time-consuming to determine what these proteins do, and how they do it. Modern automated techniques are revolutionizing many aspects of biology, for example, supporting

extremely fast gene sequencing and massively parallel gene expression testing. Protein structure determination, however, remains a long, hard, and expensive task. High-throughput, automated, algorithmic methods are required to apply modern techniques such as computer-aided drug design on a much larger scale. New algorithms in structural biology and molecular biophysics are advancing our long-range goal of understanding biopolymer interactions in systems of significant biochemical as well as pharmacological interest. For example, to analyze noncrystallographic symmetry in X-ray diffraction data of biopolymers, one must "recognize" a finite subgroup of $SO(3)$ (the Lie group of 3-dimensional rotations) out of a large set of molecular orientations. The problem may be reduced to clustering in $SO(3)$ modulo a finite group, and solved efficiently by "factoring" into a clustering on the unit circle followed by clustering on the 2-sphere S^2, plus some group-theoretic calculations. This yields a polynomial-time algorithm that is efficient in practice, and which recently enabled biological crystallographers to reveal the architecture of a parasite's enzyme (chapter 47), which will help researchers reduce the threat of certain diseases among those with weak immune systems.

The preceding example illustrates an algorithm for *analyzing* biological data and biological problems. However, another family of algorithms can be applied to *synthetic* problems such as protein engineering. Recently developed ensemble-based scoring and search algorithms for protein design have been applied to modify the substrate specificity of an antibiotic-producing enzyme in the nonribosomal peptide synthetase (NRPS) pathway (chapter 12). Realization of novel molecular function requires the ability to alter molecular complex formation. Enzymatic function can be altered by changing enzyme-substrate interactions via modification of an enzyme's active site. A redesigned enzyme may perform either a novel reaction on its native substrates or its native reaction on novel substrates. A number of computational approaches have been developed to address the combinatorial nature of the protein redesign problem. These approaches typically search for the global minimum energy conformation among an exponential number of protein conformations (chapter 11). New algorithms for protein redesign combine a statistical mechanics-derived ensemble-based approach to computing the binding constant with the speed and completeness of a branch-and-bound pruning algorithm. In addition, efficient deterministic approximation algorithms have been obtained, capable of approximating biophysical scoring functions to arbitrary precision. These techniques include provable ε-approximation algorithms for estimating partition functions to model binding and protein flexibility (chapter 12).

These two examples illustrate specific studies where advanced algorithms have provided leverage in biological macromolecular structure determination and in the design of enzymes. This book explains the algorithmic foundations and computational approaches underlying these, and other technical subfields in structural biology, including NMR structural biology; X-ray crystallography; design of proteins, peptides, and small molecules; analysis of protein:protein interactions; modeling of protein flexibility; protein:ligand binding; protein loops; and intrinsically disordered proteins (see table P.1).

This book is based on lectures from a course I teach at Duke University, called "Algorithms in Structural Molecular Biology and Molecular Biophysics." While some undergraduate and

Table P.1
Ten biological and biophysical topics for which algorithms are developed in this book

Topic	Chapters
NMR structural biology	1–8, 13–18, 39, 29, 31, 33–35, 43–45
X-ray crystallography	
Crystallography methodology	40, 47, 48
Analysis and use of X-ray structures	9–12, 14, 20–30, 40, 41, 49, 50.
Design of proteins, peptides, and small molecules	9–12, 25, 46, 45, 27, 28, 30, 41, 26, 49
Protein:protein interactions	9, 10, 23, 27, 28.
Protein flexibility	5, 9–12, 20–26, 30, 41, 42, 25
Protein:ligand binding	9, 10, 12, 25–29, 40, 41, 46
Protein loops	20, 22, 24
Intrinsically disordered proteins	7, 39, 42
Protein kinematics	1, 2, 9, 15–18, 13, 20–24, 30
Modular enzymes	10, 12, 27, 28, 46

graduate students may have had a course in sequence-based computational biology (for example, genomics or even systems biology), in general, the algorithmic content of such a course will not prepare them for the challenges of computational structural biology or algorithms for molecular biophysics. This book is intended to fill that gap, but is a standalone course that does not presume a formal background in computational biology.

I hope the exposition in this book, and the algorithms I describe, will be generally useful both to computer scientists and to the structural biology community, not only for the earlier examples but also in other areas, including studies of protein:ligand binding and protein redesign that are experimentally driven. For example, provable approximation algorithms encode quite general techniques for computer-assisted drug design, and for docking flexible ligands to flexible active sites. I believe there are broad potential applications of these algorithmic techniques for determining structures, modeling protein flexibility, designing proteins, and modeling the biophysical processes of binding and catalysis in protein biochemistry. In each of these topics, computational techniques are central, and the applications present intriguing problems to computer scientists who design algorithms and implement systems. The next generation of computational structural biologists will need training in geometric algorithms, provably good approximation algorithms, scientific computation, and an array of algorithmic techniques for handling noise and uncertainty in combinatorial geometry and computational biophysics. This book is designed to speed young scientists on their way to research success in this exciting endeavor.

How to Use This Book

This textbook is intended for first- or second-year graduate students, or advanced undergraduates (juniors or seniors, in the American system). It should also serve as a reference book to more

advanced students and researchers, and as a resource on algorithms important to computational structural biology and molecular biophysics. The book is organized into chapters called "lectures," each of which covers concepts essential and important to the field. Many of the primary research papers on these subjects cover difficult algorithmic problems and are dense, so my goal was to have short chapters that, when carefully read and studied, could be pondered like koans by a student, and provide the foundation for launching her into research.

Each "lecture" focuses on a key topic and is typically much shorter than a review article, covering the computational process and biophysical highlights from the original, primary papers in a compact and (I hope) thought-provoking style. An exception to the brief chapter format is provided in the "short course" in lectures 15–18, covering algorithmic and computational issues related to NMR structural biology. The "short course" allows the student to build up sufficient momentum to master the fascinating yet thorny computational problems that require a sustained attack and uninterrupted exposition. These four lectures could also provide a standalone short course for a laboratory, departmental, or center retreat, or they could be a subtheme of a longer course.

This book interleaves important themes in structural computational biology, including algorithms, NMR, design of proteins and other molecules, and macromolecular flexibility. For difficult topics, a didactic style of increasingly deeper repetition is employed (for example, for Dead-end Elimination in protein design, and residual dipolar couplings in NMR), in which a biophysical theme or algorithm is gently introduced in a brief form, and then periodically revisited in a cyclic and iteratively deeper fashion throughout the book.

A key theme of this textbook is understanding the interplay between biophysical experiments and computational algorithms. I try to emphasize the mathematical foundations of computational structural biology while balancing between algorithms and a nuanced understanding of experimental data. The book concentrates on the information content of the experiments, and the methods for design and analysis of protein structure, together with techniques for macromolecular structure determination, providing an emphasis, where possible, on provable algorithms with guarantees of soundness, completeness, and complexity bounds. In particular, rather than describing a competition between computer programs, this book tries to evaluate the strengths and weaknesses of the underlying ideas (algorithms). There are several reasons. First, I believe no one will be using the same programs in 10 years (and if we are, that would reflect poorly on the field). However, the underlying mathematical relationships between the data and the biological structures should prove enduring, warranting a characterization of the completeness, soundness, and complexity of algorithms in structural molecular biology and molecular biophysics.

This book describes the development of algorithms for a broad array of biological, biophysical, and biochemical topics in structural biology. A list of ten important topics is given in table P.1. I used several guiding principles in designing this book. First, several excellent textbooks already exist on sequence-based computational biology. There are fewer books on computational aspects of structural biology, and fewer still that focus on the interplay between advanced algorithms and raw experimental data. This book emphasizes computational biology from a structural and

biophysical point of view, and tries to place a greater emphasis on algorithms, especially provable algorithms, and the underlying mathematics. Three emerging areas are stressed because they are particularly fertile ground for research students: NMR methodology, design of proteins and other molecules, and the modeling of protein flexibility. Where possible, the focus is on combinatorially precise algorithms with provable properties. In particular, I have tried to convey the rich geometric and algebraic structure of the underlying algorithms.

NMR and X-ray Crystallography At the same time, the topics in this book are not exhaustive. They have been selected with the following intent. First, where possible, the algorithms should be nontrivial and therefore have the possibility to inspire computer scientists to work on algorithms for structural molecular biology. The exposition, however, should be at such a level that biochemistry students will be moved to learn about useful structural algorithms for their work. In order for the course to be feasible to complete within one year, I have made some choices about what to cover. Even though the majority of the protein and DNA structures in the protein data bank (PDB) have been determined by X-ray crystallography, this book gives more emphasis to algorithms for solution-state NMR. One reason is that, while both techniques are undergoing rapid technology development, NMR is currently less automated than crystallography and therefore there are probably more algorithmic opportunities for computational scientists to make their mark in developing new algorithms to process, analyze, and make inferences from the new biophysical experiments being developed, on a seemingly weekly basis, by NMR spin gymnasts. This having been said, some fascinating computational problems arising in experimental X-ray crystallography of proteins are discussed in chapters 40, 47, and 48, and, of course, the analysis and use (for example, in design, docking, and the modeling of protein flexibility) of these crystal structures are covered in chapters 9–12, 14, 20–30, 40, 41, 49, and 50.

Design of Proteins, Peptides, and Small Molecules The design of proteins in general and enzymes in particular represents one of the most important problems in biochemistry and synthetic biology. An example of the computational and experimental challenges in protein design is given at the beginning of this Preface. Similarly, design, identification, and discovery of small molecule inhibitors are of central biochemical and pharmacological importance. Therefore, many chapters of this book are devoted to the design of proteins, peptides, and small molecules: chapters 9–12, 25, 46, 45, 27, 28, 30, 41, 26, and 49.

Protein Flexibility In the early days of structural biology, it was an experimental challenge even to obtain a single structural model of a biological macromolecule such as a protein. Furthermore, computational techniques were limited both in terms of algorithmic sophistication and processor speed, and therefore could only deal with a single three-dimensional model. More recently, experimental techniques have yielded models of proteins that encompass some of the dynamics and mobility of proteins and nucleic acids in solution-state physiological conditions. These range from ensemble representations (sets of conformations) to more direct measurements of

mobility such as motion tensors (from NMR relaxation experiments or residual dipolar couplings). Moreover, computer processor speeds have increased dramatically, and massively parallel distributed cluster or cloud computations have become routine. Algorithms have improved in scope and sophistication, and now are beginning to handle some of the challenges of modeling macromolecular flexibility. Therefore, algorithms for modeling and computing with representations of protein flexibility are emphasized in this book, since they will be of increasing importance in the future of structural biology: chapters 5, 9–12, 20–26, 30, 41, 42, and 25.

Protein:Ligand Binding Biological macromolecules such as proteins do not exist in isolation. They move, flex, and dynamically bind to partner ligands. Enzymes must selectively bind to substrates and cofactors in order to perform their reaction chemistry. Protein:protein interactions are ubiquitous for cell signaling, regulation, transcription, and translation. Algorithms to predict the functional interactions of proteins with other molecules must be able to model the affinity of protein:ligand binding. These algorithms are developed and discussed in chapters 9, 10, 12, 23, 25–29, 40, 41, and 46.

Stochastic and Heuristic Techniques I do not explicitly cover in separate chapters the stochastic techniques of Monte Carlo, simulated annealing, Metropolis algorithms, or genetic algorithms. These techniques are well-treated in both other books and excellent reviews, and therefore they are discussed as an ingredient of certain lectures (e.g., 6, 9, 10, 11, 20, 22–26, 30, 31, 41, 43, 44, 45, and 49) rather than as the main course. This also fits well with our emphasis on provable algorithms, since these stochastic techniques are heuristic, typically admitting no provable guarantees. In previous reviews and textbooks, provable algorithms are less generously covered, and therefore they are given more attention in this book. Again, this choice is intentional, and is motivated to lure young computer scientists into the field.

Protein Kinematics On the other hand, there are some rigorous and well-developed areas that are mentioned only briefly in this book, because excellent textbooks and reviews of these techniques already exist. These areas include molecular dynamics simulations, geometric hashing, protein threading, and the Vereshchagin-Featherstone linear-time recursive Lagrangian dynamics algorithm. These four topics have also been successfully integrated into many best-practices algorithms. With regard to algorithms for kinematics, I take a middle way, and provide material on the interplay between geometry, algebra, and experimental restraints in defining protein kinematics (lectures 1, 2, 9, 15–18, 13, 20–24, and 30). This having been said, my book is not a general text on kinematics, a subject that is also well-represented by other authors. Finally, it must be mentioned that for many of the topics in this book there is a large literature that we do not cover. For example, an entire course could be taught on any one of the following subjects: normal mode analysis, protein loops, enzyme kinetics, Markov random fields, energy minimization, computational topology, protein design, singular value decomposition and its applications, and the structural bioinformatics of rotamer and small molecule libraries. Though each of these

is discussed briefly, to fit within fifty chapters, I had to choose the material I thought was most accessible for my student audience. Consequently, a lot of thoughtful and significant papers are not mentioned. I hope that engaged students will pursue their interests further by doing a detailed search on Medline for the most recent papers complementing the brief chapters in this book.

Background and Prerequisites Students taking a course based on this book should have a background in algorithms (for example, an advanced undergraduate or early graduate course on algorithms) plus some exposure to the notions of NP-completeness. The latter may be waived for open-minded students with a mathematics or physics background, and what is sometimes called "mathematical sophistication." It is also helpful to have a background in basic biochemistry or to be taking a structural biochemistry course concurrently. For students in the life sciences, this textbook and a course based thereupon should be valuable if they have some background in computer science and are willing to read more. The book is organized as a year-long lecture course, to be taught in the order I provide. As a two-semester or two-quarter course, it allows time for presentations of student projects during the last two weeks of the second semester. It is recommended that student projects should consist of (a) choosing an algorithm covered in one of the lectures, (b) implementing it from scratch, and (c) trying it on real data obtainable from one of the data or structure repositories (e.g., PDB, BMRB, etc.). When I use this book in teaching, I have computer scientists, computational biologists, and biochemists in the class. Therefore, being somewhat relaxed about prerequisites is recommended to include different backgrounds and have a more lively discussion. For projects and assignments, it is useful to pair students with a "dry" (computational) background with those from a wet lab (experimental) background.

Courses of Different Lengths

There are alternative paths through this book for a shorter, one-semester or one-quarter class. An excellent one-semester class would cover only lectures 1–5, 8, 9, 11, 12, 15–18, 20–23, 25, 31, 32, 37, 38, 45, 47, and 50. As mentioned earlier, a 3- to 4-day short course on NMR methodology could be taught stand alone based on lectures 15–18. Alternative paths through the book could emphasize different themes. One path for one semester would emphasize algorithmic and computational approaches in NMR structural biology and cover lectures 1–8, 13–18, 39, 29, 31, 33–35, and 43–45. Another route would emphasize design (especially protein design) covering lectures 9, 10, 11, 12, 25, 46, 45, 27, 28, 30, 41, 26, and 49, in that order.

A third path would emphasize the general theme of protein flexibility, covering lectures 5, 9–12, 20–26, 30, 41, 42, and 25. Last, a running example is given, of particular biological systems, from the family of enzymes known as nonribosomal peptide synthetases (NRPS). The reason for this is fourfold. First, it is useful to have an example where particular residues, active sites, and substrates can be named and visualized. Second, although we envision all these algorithms as quite general, in practice most algorithms still have some interplay or dependence on the kind of systems they have been developed and demonstrated on. We have not achieved a complete

decoupling or genericity of the algorithms from the underlying data and structures. Therefore, describing particular biological systems mirrors the reality that they are usually interleaved with the algorithms when we read about them in the literature. Third, vacationing from mathematical algorithms into the countryside of the biochemistry can be a comforting and inclusive excursion for students with a life science background. Fourth, and perhaps most important, this system of modular enzymes presents fascinating opportunities for computational design and therefore may inspire a new generation of students to tackle them in their research. At any rate, a short subtheme or path covering the NRPS (and related) enzymes is given in the sequence of lectures 10, 12, 27, 28, 46. If my recommended sequence for a year-long or semester-long course does not fit the instructor's needs, the NMR short course and the four paths above (namely, NMR, design, protein flexibility, and NRPS) can be mixed and matched to create a course that is tailored for a particular occasion or purpose.

The preceding discussion about alternative paths and subthemes may give the mistaken impression that these are the only topics this book covers in depth. To the contrary, as can be seen from table P. 1 or by browsing the table of contents, we cover a wide range of topics in computational structural biology. For example, I hope instructors will note the two chapters on computational mass spectometry (19, 43), an increasingly valuable technique in biophysics and proteomics. However, the paths and subthemes of NMR, design, protein flexibility, and NRPS are distinguished in that the later lectures in each path presume some knowledge of the earlier lectures in that track. However, most of the lectures outside these tracks can be read independently. Therefore, they can also serve as reference chapters for researchers in the field, or be put together in an order that suits an individual instructor for a particular course. In particular, a nice section of a course on protein:ligand binding could be based on chapters 9, 10, 12, 25–29, 40, 41, and 46, read in that order.

Finally, the chapter on belief propagation (25) could serve as a general introduction to Markov Random Fields and factor graphs; the chapters on distance geometry (1, 31, 32, 36) or computational topology (50) would be at home in an algorithms or computational geometry course; and any of these could be followed by the chapters on graph cuts (37–38). That being said, these lectures are all motivated by the underlying biophysics and hence more trenchant when nestled in the context of macromolecular structural biochemistry within the logical sequence of this book.