

PREFACE

What's this book about?

The computer is the most powerful general-purpose tool available to biologists.

In part, this is due to the continuing rapid growth of biological data. For example, at the time of writing, the GenBank database had over 100 *million* genetic sequences with over 100 *billion* DNA characters. Among the contents of that database are genes from many organisms, annotated with what's known about their function.

Imagine that you're studying a bacterium and wish to understand what causes it to be infectious. One promising approach is to identify genes in the bacterium and compare these to known genes in GenBank. If you're able to find similar genes whose function is known, it will tell you a great deal about the role of the genes in your bacterium. This approach represents a computational challenge, and is, in fact, the topic of Part I of this book.

But searching enormous databases is not the only reason that computers are so useful to biologists. Many biological problems have a large number of different possible solutions and only a computer – programmed with carefully designed computational recipes or “algorithms” – has any chance of finding the right one. For example, biological molecules such as proteins and RNA fold into complex shapes that strongly impact their function. Computational techniques have been developed to predict how these molecules fold. Such techniques help us understand how proteins and RNA work and can even help us design new molecules to treat disease.

Simply put, computing is revolutionizing the practice of biology.

In order to fully appreciate and exploit the power of computation, biologists must be trained to “think” computationally. In practical terms, this means understanding fundamental computing concepts that recur in many applications *and* being able to write programs.

Why? Consider, for example, that there are well over 400 different software packages for phylogenetics (the study of the evolutionary relationships among organisms). An increasing level of computational sophistication is needed to select the appropriate software for a given application, use it correctly, and understand its

abilities and limitations. This is true not just of phylogenetics, but also for many other areas of biology.

But using existing software will not be enough. The number and variety of computational problems that arise in biology are rapidly outpacing the functionality of software tools. Eventually, most biologists are likely to encounter problems that cannot be solved with existing software. Therefore, it is imperative for biologists to have the ability to write their own programs.

This book seeks to provide biology students with both an exposure to major computational ideas *and* practical programming skills. It requires no specific biology or computer science background. It is designed as a first-year college-level course and has been taught to that audience at Harvey Mudd College since 2009. The authors hereby acknowledge the generous grant support of HHMI for the development of that course.

In contrast to a typical introductory bioinformatics book, this book emphasizes programming over the use of existing software. By the time you've completed this book, and the online homework problems, you should feel comfortable writing programs for a wide array of applications in biology and beyond. You'll also have an understanding of computational ideas like "heuristics," "memoization" and "dynamic programming," "NP-completeness," and others, that will allow you to understand and compare the technical aspects of existing bioinformatics tools.

This book begins with Chapter 0, which offers a first introduction to the Python programming language. The rest of the book is organized into four parts, each comprising several chapters. Each part begins with a "large" biological question and the chapters in that part provide the computational and programming tools to answer that question.

At the end of each chapter – and sometimes even within a chapter – you'll see a question icon pointing you to one or more recommended problems. You'll find those problems at the url:

www.cs.hmc.edu/CFB

Let's get started!

