

Preface

"...we seem to agree that [bioinformatics is] an interdisciplinary field, requiring skills in computer science, molecular biology, statistics, mathematics, and more. I'm not qualified in any of these fields...though I spend most of my time writing software, developing algorithms, and deriving equations..."

— Sean Eddy¹

One of the challenges of writing a textbook in an interdisciplinary field as fluid and rapidly evolving as bioinformatics is that no one is an expert in all aspects of the field. We all come to the table with different backgrounds, a collection of skills accrued along the way, and a set of ideas of what to study and how to go about doing so. To be successful you need to communicate your ideas to people who share neither the background nor your vocabulary, and to do so in a way that fires the imagination. That's what we have tried to do in this book. We are a disparate group: formal training in mathematics and computer science for two of us; formal training in biology for the other two; all of us using, developing and evaluating new computational methods for the analysis of biological problems. What gets us out of bed in the morning is the idea that we can design an algorithm or computational method that will help us better understand the miracle of life. Along the way, we discovered that together we could create a textbook that blends computing and biology in an engaging and unique way.

In fact, we believe this book is as close to a stand-alone text as possible in the field. The book is aimed at those with a computing background, but it contains enough background from both the biological and computing angles that students from either discipline should feel comfortable starting with this text. Of course, no text can cover everything in equal depth, so when push comes to shove, we slide to the computing end of the scale rather than the biology end. Thus, we assume that students have already acquired about a year's worth of programming experience and are conversant with Java or a similar object-oriented programming language.

To help keep students focused on material that is often complex and detailed, we used one long-running example throughout the text. Specifically, we provided worked examples and discussed the ways in which computing approaches have elucidated key aspects of the biology of the human immunodeficiency virus (HIV), the virus that causes AIDS. Threading a single example through many topics is meant to enable computing students to ground themselves as they explore the complexities of biology. In turn, the HIV example gives biology students a chance to return to a familiar place even as advanced computing concepts are introduced.

¹ Eddy, Sean. (2005). "Ante disciplinary Science," *PLoS Computational Biology*. 1(1):e6.

We made a conscious decision not to talk about the application of common bioinformatics tools, although we highlighted interesting methods that are embedded in well-known programs. This book is not a "how-to" guide to bioinformatics, and its intended audience is not computer scientists hoping to pick up a little biology or biologists hoping to get a few computing clues. Rather, we hoped to engage the next generation of scientists. We wanted to inspire individuals to design the breakthrough algorithms and computational approaches that will provide novel and valuable insights into biological phenomena. Ambitious as this is, we have learned from our experiences teaching undergraduates that if you demand the world of them, some will produce amazing results. Indeed, we wrote this book for just those undergraduates who over the years have surprised us with their ingenuity, tenacity and enterprising approach to the field. This is a book aimed at all the undergraduates and possibly first year graduate students who are just itching for the chance to try out their skills on a whole new set of problems.

We purposely took a conversational tone in the text, keeping things relatively lighthearted even as we engaged in rigorous analysis. We hoped in doing so to entice students into learning and thinking.

We did not think in such lofty terms when we began this process several years ago. We started at a much more prosaic level: Anne Haake and Rhys Price Jones were offering a pair of courses titled "Introduction to Bioinformatics Computing" and "Advanced Bioinformatics Computing" for undergraduates at the Rochester Institute of Technology (RIT). None of the textbooks on the market had the right blend of computing and biology for the courses. We identified many textbooks that were merely procedural guides describing how existing computational tools could be applied to specific biology problems. Another set of textbooks was so overwhelmingly focused on the algorithmic aspects that they lost sight of the biology that motivated them. We wanted to achieve that happy medium: a textbook that would emphasize the theoretical underpinnings and logic of the algorithms while also highlighting the biological knowledge that drives the algorithm development in a unique direction. Since no one else was writing the textbook we wanted, we wrote our own. The fruits of those labors reside in your hands now.

We will not tell you how to use this book because we think there are many ways to use it. What we do suggest is that you feel free to pick those chapters that most suit your tastes and the needs of your students. For example, if you or your students need a little extra biology background, you will find that Chapters 2 and 3 provide a very good primer on basic molecular biology. If, on the other hand, your students need a gentler introduction to some of the algorithms, chapters 3, 4, and 5 are designed to help them understand the spirit of bioinformatics algorithms. These chapters do, however, expect a familiarity with written explanations of algorithmic ideas, such as would be encountered in most introductory computer science courses at the college level.

You can certainly work through all the chapters in the book over the course of a semester or year-long course depending on the level of detail you wish to pursue. You can customize students' experience by selecting from the exercises included in each chapter. Many of these exercises require students to develop code

and programs to accomplish specific tasks. Most of these tasks have a biological basis or are motivated by a specific biological phenomenon. For a faster paced, less involved course, you might omit some of the more detailed programming exercises. Additional instructional material and solutions to the exercises will be made available at www.mhhe.com/gopal.

Chapter 9 encourages you to give your students free rein to explore some fascinating new areas of bioinformatics where computing challenges loom large – areas in which the extant solutions come in an intriguing variety and the open questions are just as numerous. Rather than try to cover all the details of ongoing research, we leave the door open for innovative students to investigate the issues for themselves, either on their own or in small groups. This provides an important pedagogical tool: the motivation and context within which students can become independent investigators in their own right.

In the end, we want this book to be a gateway to new and exciting discoveries. We invite you to explore, to investigate and then to roam free through the field. If our experiences are any indication, what you produce from these explorations will surprise even you. We hope you will enjoy the journey as much as we have.

Shuba Gopal

Rochester, NY, May 2008

Anne Haake

Rochester NY, May 2008

Rhys Price Jones

Washington, D.C., May 2008

Paul Tymann

Rochester, NY, May 2008