Don’t Fear the Command Line!

Practical Computing for Biologists

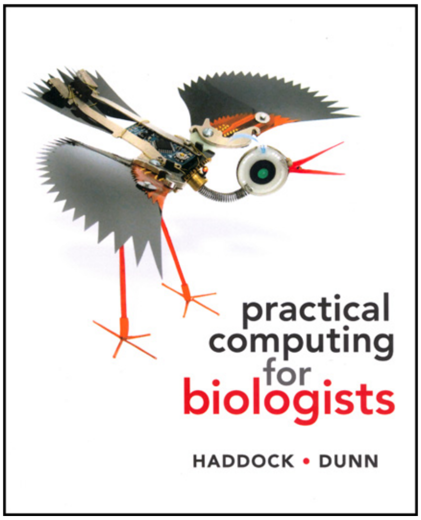
Authors: Steven Haddock and Casey Dunn
Sunderland, MA, USA: Sinauer Associates, Inc. (2010). 538 pp. $59.95.

Although basic computing skills are routine for most biologists, most of us still struggle with more sophisticated tasks, beyond the “out of the box” solutions. Unfortunately, day-to-day lab work increasingly involves more advanced computing challenges. For example, you may find yourself wanting to merge several gene expression microarray files, convert them into a format compatible with the clustering software, perform statistical analysis on all the resulting clusters, and plot the results. Worse still, you might need to organize your laboratory’s microarray studies into a single local database and apply the above analysis pipeline to all of them. How does a bench biologist solve problems like these?

A new textbook, *Practical Computing for Biologists*, by Steven Haddock of the Monterey Bay Aquarium Research Institute and the University of California, Santa Cruz and Casey Dunn of Brown University aims to teach biology researchers the computing skills necessary in such situations. The authors describe themselves as “biologists who also happen to have backgrounds in computing,” and they provide a problem-oriented approach to addressing data analysis and presentation challenges in modern biology.

The book covers a wide range of subjects that truly justifies the title of “practical computing.” In addition to the usual programming-related topics, it also includes a thorough introduction to the programming environment, approaches to combining different programs together, a description of the basic text manipulation tools such as regular expressions, and even an introduction to dealing with digital art and images. As such the book is great value for the money, being at least three books in one. Readers will benefit from the breadth of topics covered, from Python programming and image manipulation, to databases and even electronic circuits. The most dedicated can even start learning about more advanced topics such as relational databases, though the single chapter covers only the basics of setting up and managing MySQL. One potential omission is the lack of web-related topics, which could perhaps make an interesting addition to the second edition.

The textbook’s broad scope is both an advantage and a shortcoming. On the positive side, any student who learns the full content will not only be versed in Python programming and image manipulation but also have a rudimentary understanding of databases and circuits. On the negative side, some sections might be too advanced for the more casual reader. In addition, the overall organization of the book is sometimes puzzling, making it harder to identify the “must-read” parts. For example, chapter 20 provides useful tips on how to connect to remote computers through secure connections and how to control programs on your machine. These skills would be helpful to have before learning to program in chapters 7 through 13 because most modern scientific computing is performed on remote servers and clusters. In addition, the section that focuses on combining programs and methods (Part IV) also includes a lone chapter on relational databases—advanced material that perhaps is not necessary for most readers.



However, despite these shortcomings, this is an excellent textbook, especially its introduction to programming in Parts I–IV. The book is well written in a lively style that is grounded in many concrete examples. Logical formatting combined with an excellent use of color and icons make it easy to follow, even for beginners. In addition, the clear explanations ensure that students are not simply retyping example programs but are actually learning how to solve real-world biological problems with programming. The choice of Python as the programming language is astute, given its relative lack of complexity, its broad utility, and the availability of robust scientific and biology-specific libraries.

The beginning of the book (Part I) covers basic text manipulation, starting with installing and learning how to use a text editor. The authors chose Mac OS X as the setting for all of the book’s examples and specifically focus on TextWrangler, a free editor that only runs under this operating system. Although such a specific setting makes it easier for students to follow exercises, the choice of one particular operating system and a text editor that is fully tied to it seems like a limitation, especially given the proliferation of freely available text editors for every common platform. The rest of this section provides background for learning how to use regular expressions for search and replace-type operations on various text files, a very useful skill for analyzing biology datasets.

The “meat” of the book is in Parts II–IV, which focus on programming both through command-line operations and in Python scripts. This section also cultivates an understanding of how to combine multiple methods together to address more complex tasks. The authors deserve praise here, as the book strikes an excellent balance between teaching relatively advanced programming skills and remaining utterly practical and easy to follow. For example, a whole chapter is devoted to explaining debugging techniques, an area that many biologist programmers are never taught, leading to many frustrating hours that force them either to rediscover debugging on their own or to give up programming entirely. The book also provides many practical pointers that can help streamline future programming efforts, such as learning how to best organize data in spreadsheets to simplify down-the-line processing and analysis.

The remaining sections of the book are a mix, including sections on creating and working with vector art, manipulating images, and basic electronics, while others focus on practical topics such as installing and troubleshooting software and working on remote computers. The appendices are short but very useful, including a nice reference section for topics covered earlier in the book (Python, shell and SQL commands, and regular expression terms) as well as a short guide to working with Windows and Linux operating systems. A nice and unusual teaching tool can be found in Appendix 5, where the same template program is presented in several different programming languages.

The many examples presented throughout the book provide a solid foundation for the programming concepts and make the book easier to digest. The book is billed as ‘‘standing on its own,’’ and indeed, it could be a good self-study guide for students and professionals. Parts I–IV could also serve as study material for a ‘‘Programming for Biologists’’ course at either the undergraduate or graduate level, although in this case, the book would benefit from the addition of exercises and test questions at the ends of the chapters.

Overall, *Practical Computing for Biologists* is a good choice and great value for a textbook. As a reference manual, there may be better options whose organization is more suited for fast information retrieval (such as the O'Reilly series). However, *Practical Computing for Biologists* provides a clear and sophisticated background in programming for the experimental scientist and lays the foundations for more advanced topics that many biologists are likely to find increasingly useful.

Olga G. Troyanskaya[1,*]
[1]Department of Computer Science and Lewis-Sigler Institute for Integrative Genomics, Princeton University, Princeton, NJ 08544, USA
*Correspondence: ogt@cs.princeton.edu

DOI 10.1016/j.cell.2011.02.042