Using R for Computer Simulation and Data Analysis in Biochemistry, Molecular Biology, and Biophysics

Victor A. Bloomfield

Department of Biochemistry, Molecular Biology, and Biophysics
University of Minnesota
321 Church St. SE
Minneapolis, Minnesota 55455
victor@umn.edu

Abstract. Modern biology has become a much more quantitative science, so there is a need to teach a quantitative approach to students. I have developed a course that teaches students some approaches to constructing computational models of biological mechanisms, both deterministic and with some elements of randomness; learning how concepts of probability can help to understand important features of DNA sequences; and applying a useful set of statistical methods to analysis of experimental data. The free, open-source, cross-platform program R serves well as the computer tool for the course, because of its high-level capabilities, excellent graphics, superb statistical capabilities, extensive contributed packages, and active development in bioinformatics.

1 Introduction

1.1 The Need for a More Quantitative Bbiology

The Executive Summary of the influential 2003 report from the National Academy of Sciences, "BIO 2010: Transforming Undergraduate Education for Future Research Biologists" [1], begins

> The interplay of the recombinant DNA, instrumentation, and digital revolutions has profoundly transformed biological research. The confluence of these three innovations has led to important discoveries, such as the mapping of the human genome. How biologists design, perform, and analyze experiments is changing swiftly. Biological concepts and models are becoming more quantitative, and biological research has become critically dependent on concepts and methods drawn from other scientific disciplines. The connections between the biological sciences and the physical sciences, mathematics, and computer science are rapidly becoming deeper and more extensive.

Quantitative approaches have become particularly prominent in the large-scale approaches of systems biology and its associated high-throughput techniques: bioinformatics, genomics, proteomics, metabolomics, cellomics, etc.

G. Allen et al. (Eds.): ICCS 2009, Part II, LNCS 5545, pp. 25–33, 2009.

High levels of quantitation are also needed in some of the more biophysically-oriented aspects of biochemistry, molecular and cellular biology, physiology, pharmacology, and neuroscience.

The increasing use of quantitation at the frontiers of modern biology requires that students learn some basic quantitative methods at an early stage, so that they can build on them as their careers develop. To deal with realistic biological problems, these quantitative methods need to go beyond those taught in standard courses in calculus and the elements of differential equations and linear algebra—courses based mainly on analytical approaches—to encompass appropriate numerical and computational techniques. The types of realistic biological problems that contemporary science is facing are generally too large and complex to yield to analytical approaches, and specific numerical answers are usually desired, so in many cases it makes sense to go directly to computational rather than analytical mathematical answers.

Modern molecular and cellular biology also demands increasingly sophisticated use of statistics, a demand difficult to meet when many life science students don't take even an elementary statistics course.

1.2 Computer vs. Analytical Tools

To add significant instruction in computational and statistical methods to an already overcrowded biology curriculum poses a challenge. Fortunately, modern computer tools, running on ordinary personal computers, enable very sophisticated analyses without requiring much analytical or programming knowledge or effort. In essence, this is a "black box" approach to quantitative biology; but I contend that using a set of black boxes is better than not using quantitative tools at all when they would substantially enhance the results of biological investigations. The challenge, then, is to make students—and more mature scientists—aware of the appropriate black boxes, their capabilities, and the steps needed to access those capabilities. With modern computer tools, this requires only a small amount of programming and an even smaller amount of analytical manipulation.

1.3 The Choice of R as the Computational Tool

R is a free software environment for computer programming, statistical computing, and graphics. The R web site [2] emphasizes statistical computing and graphics, which it does superlatively well; but R is also a very capable environment for general numerical computer programming.

R has many characteristics that make it a good choice on which to build quantitative expertise in the biochemical sciences. Its capabilities are similar to those of excellent and widely-used but expensive commercial programs. It runs on Mac OS, Windows, and various Linux and Unix platforms. It is free, open-source, and undergoing continual (but not excessive) development and maintenance. It is an evolving but stable platform that will remain reliable for many years. It has a wide variety of useful built-in functions and packages, and can be readily extended with standard programming techniques. It has excellent graphics.

If needed for large, computationally demanding projects, R can be used to interface with other, speedier but less convenient programming languages. Once its (fairly simple) syntax is learned, it is easier and more efficient than a spreadsheet. It has many sample datasets, which help with learning to use the program. It is widely used in statistics, and is increasingly used in biological applications, most notably the Bioconductor project [3].

Because of these characteristics, R can serve students as their basic quantitative, statistical, and graphics tool as they develop their careers.

2 Syllabus for the Course

The course is designed for one semester. It is divided into four main parts, with 14 "modules" corresponding to 14 weeks of the course and chapters of the accompanying textbook [4]. Two additional weeks are allocated for midterm and final examinations.

2.1 Part 1: The Basics of R

Calculating. In this and the next module, we begin with the most basic aspects of R: installing it, checking the installation by demonstrating some of its impressive graphics, and showing how it can be used as a powerful calculator with vector and matrix capabilities.

Plotting. An important part of scientific computing and data analysis is graphical visualization, an area in which R is very strong. R has many specialized graph types, some of which are explored later in the course. However, for many scientific purposes just a few types will suffice, especially graphs for data, functions, and histograms. We first give simple examples, and then show how they can be customized.

Built-in functions, user-defined functions, and programming. The base installation of R has many built-in functions, including `sort` and `order` to arrange vectors or arrays in ascending or descending order; all the standard trigonometric and hyperbolic functions `log`(base e), `log10`, `exp`, `sqrt`, `abs`, etc.; and more sophisticated mathematical functions such as `factorial`, `gamma`, `bessel`, `fft` (Fourier transform), etc. Additional mathematical functions, the orthogonal polynomials used in mathematical physics and chemistry, are available in the contributed package `orthopolynom`, available through the CRAN web site [2]. The functions `uniroot` and `polyroot` are used to solve for the zeros of general functions and polynomials, respectively. In addition to those mathematical functions, R has numerous others that are useful to scientists, including sorting, splines, and sampling. We also show how to define new functions, with examples of Gaussian functions and pH-titration curves.

Programs in R, as in most computer languages, typically consist of a few standard types of operations: assigning the values of variables and evaluating expressions involving those variables; conditional execution, in which different

sequences of statements are executed depending on whether a expression is true or false; and repetition or looping, in which an action is performed repeatedly until some condition is met. R is generally thought of as a programming language for statisticians, but it has the capabilities needed for the sort of numerical analysis done in most sorts of scientific work. The module concludes with some of the most common examples: finding the roots of polynomials or other functions, solving systems of linear and nonlinear equations, and numerical integration and differentiation.

Other important tasks, such as numerically solving differential equations, fitting data to linear or nonlinear equations, and finding periodicities in data with spectral analysis and Fourier transforms, are introduced in subsequent chapters.

Data and packages. Up to this point we have mainly dealt with how to use R for calculating and graphing. In programming for scientific work we also generally need to get data from various sources, transform it, and save it for later use. We also will often wish to augment the built-in capabilities of R with more specialized resources. Many such resources are available as contributed packages from the CRAN web site [2]. This module deals with those two important topics: handling data and adding packages.

2.2 Part 2: Simulation of Biological Processes

Equilibrium and steady state calculations. Much of biochemistry, molecular biology, and biophysics deals with the the equilibrium and dynamics of biochemical reactions. In this module we focus on two important types of time-independent processes: ligand binding and steady-state enzyme kinetics. These serve as test beds for showing how to use the plotting and data analysis capabilities of R.

Differential equations and reaction kinetics. In this module we show how to numerically solve the kinetic rate equations of the sort that describe biochemical metabolism, microbial growth, and similar biological phenomena. These systems of ordinary differential equations describe the change of concentrations or numbers of organisms as a function of time.

Population dynamics: competition, predation, and infection. Populat ions—whether of organisms, cells, or viruses—are of central importance in biology. In this module we consider some of the basic models of population dynamics: competition of different species for resources, predation of one species upon another, and transitions of parts of a population between different states (e.g., susceptible, infected, resistant, dead) in epidemics.

Diffusion and transport. The movement of biological molecules in cells or in lab experiments gives useful insight into their sizes, associations, and mechanisms of transport to functional locations. The movement may be random diffusion (Brownian motion), it may be in response to some driving force, or both. Familiar driving forces in the lab are electrophoretic and centrifugal fields. In the

cell, active transport and transport by cytoskeletal fibers are important mechanisms. Related situations arise in drug delivery, where the flow of drug from one compartment of the body to another can be treated by diffusion and transport models. Diffusion may be coupled with reaction as discussed in a section on regulation of morphogenesis. In this module we develop simple simulations for some of these processes. These simulations involve an introduction to the solution of partial differential equations with both space and time as independent variables.

Regulation and control of metabolism. Metabolism involves not just single biochemical reactions, but coordinated networks of reactions. These networks are usually remarkably well-regulated, keeping close to a set-point, a steady state or dynamic equilibrium in most healthy organisms. Substantial deviation from that set-point may betoken disease or some other extraordinary circumstance. On the other hand, biotechnologists may want to manipulate an organism to overproduce a desirable product, controlling its metabolism to deviate from the normal set-point. In this module we examine these issues of regulation and control by simulating the behavior of networks of enzymatic reactions.

Models of regulation. This module considers models of regulation in three different types of biological processes: transcription, response to chemotactic signals, and patterning of morphogens in cellular development. These are each huge topics, and we attempt only to present some introductory but instructive examples that are amenable to numerical simulation. An important theme is *robustness*, the ability of a system to maintain suitable functioning in the face of variations, both temporal and cell-to-cell, of biochemical parameters.

2.3 Part 3: Probability and Sequence Analysis

Probability and population genetics. Up to this point we have mainly treated deterministic processes, although we have showed how to fit noisy data and to model stochastic chemical reactions. In fact, most biological data are intrinsically noisy, or random, due to the underlying nature of the process (e.g., mutation or genetic recombination), especially when combined with the often small numbers of "individuals" in many experiments. This module discusses basic concepts of randomness and probability, and shows how these concepts may be applied in a variety of situations, concluding with a brief introduction to population genetics.

DNA sequence analysis. In this module we introduce some of the elementary concepts for analyzing DNA sequences in terms of "words" of length 1 (bases), 2 (base pairs), 3 (triplets, such as codons), restriction sites, etc. The analysis uses the basic probability concepts from the previous module.

2.4 Part 4: Statistical Analysis in Molecular and Cellular Biology

Statistical Analysis of Data. Molecular biologists and biophysicists have a lot of data to analyze, and R has a lot of tools to help with the analysis.

We consider three major topics: summary statistics for a single group of data, statistical comparison of two samples, and analysis of spectral data.

Microarrays. DNA microarrays are one of the key new technologies in biology. They are used to measure changes in gene expression levels, to detect single nucleotide polymorphisms (SNPs) that may be indicators of susceptibility to disease or useful in forensic analysis, to compare genome content of different cells or closely related organisms, and to detect alternative splicing in DNA transcription. A microarray may contain ten thousand or more spots, and therefore can carry out thousands of comparative genetic analyses at once. This enormous amount of information can provide great insight into genetic regulatory processes, but it also poses great challenges to data quality and adequate statistical analysis. This module provides a brief introduction to these issues.

3 Experience Teaching the Course

3.1 Learning and Using R

At the beginning of the course, students are told to download and install R from the CRAN (Comprehensive R Archive Network) web site [2]. To my pleasant surprise, in two offerings of the course to 28 students, none has had any trouble installing R regardless of their operating system (Mac OS, Windows, or Linux). An immediate demonstration of some of R's capabilities is obtained by running the graphics demonstration `demo(graphics)`.

The syntax of R is relatively simple and students have little trouble with the basics. The R program, as installed on the students' computers, has an extensive Help facility accessed from the menu bar. "An Introduction to R" and "R Data Import/Export" are likely to be useful as they begin learning the language. Each function has a help page, with definitions of the inputs, outputs, and options, and one or more examples of usage. These examples, however, are often terse and technical rather than readily tutorial.

A problem with the R help system is that you generally have to know the exact term being searched for, since the help system searches a pre-established index rather than the full text. For example, trying to learn about correlation analysis by typing `help(correlation)` or `?correlation` yields "Help topic not found". `?corr` gives the same result. Finally, `"cor"` brings up the desired help page. The functions `apropos` and `help.search` may (or may not) be useful in such cases. Two aids to finding online help about R topics are *RSeek* [5] and *Search the R Statistical Language* [6].

There are numerous online sites devoted to R. A particularly useful one for rapid reference is *R & BioConductor Manual* by Girke [7]. Another handy resource is the on-line *R Reference Card* by Short [8].

3.2 Useful Books

Most of the books that teach how to use R (or its progenitor S or commercial sibling S+) do so in the context of its use as a program for doing statistics.

Statistics is only one of the foci of the course, but the books by Dalgaard [9] and Verzani [10] provide useful introductions to R.

More advanced books that use R in a biological context, especially in bioinformatics, include those by Deonier et al [11], Gentleman [12], Paradis [13], Gentleman et al [14], and Hahne et al [15]. The book *Stochastic Modelling for Systems Biology* by Wilkinson [16] uses some R code in its treatment of systems biology.

In my development of this course, I have drawn heavily on *Computer Simulation in Biology: A BASIC Introduction*, by R.E. Keen and J.D. Spain (1992) [17]. This book, which appears to be out of print, uses BASIC, an earlier and much less capable computer language than R; but it has a good selection of topics and computer simulation examples for an introductory course. Of the many recent books on mathematical and computational biology, the two that fall closest to my approach, in their selection of topics and in emphasizing computational rather than analytical approaches, are those by Fall et al [18] and Allman and Rhodes [19].

3.3 Student Reaction

The course is intended for advanced undergraduates and beginning graduate students who have had basic instruction in biochemistry and calculus-level mathematics. In the two offerings thus far, the students have been a mix of senior undergraduates (most majoring in biochemistry) and beginning graduate students (the majority in masters programs in biology or microbial engineering). A few others have come from other disciplines such as computer science and chemical engineering. Neither group is very strong in analytical mathematics beyond basic calculus and linear algebra.

This diversity of backgrounds means that some students are stronger in the biological sciences, and others in mathematics and computers. Students who have had previous programming experience, but little biology, have performed better than those with a lot of biology but not much computer background. The biology students who have had the most difficulty are those whose quantitative backgrounds and interests are not strong.

Preparation in basic quantitative biochemistry (pH, equilibrium, reaction kinetics) is not strong, despite prerequisites. A number of students had particular trouble with chemical equilibrium calculations, material to which they should have been exposed in several previous courses. However, this material is notoriously difficult for mathematically-challenged students, a description that applies to many life science majors. It also appears that some of the practical implementation of equilibrium ideas, such as using difference spectroscopy to measure the relative amounts of species in a reacting mixture, are not adequately taught in prerequisite courses.

In a midterm evaluation, most of the students indicated they had a good understanding of what was expected of them, and that the combination of online lecture notes (the chapters of the book) and comments through the listserv were adequately clear. Some would have liked more illustrations or examples.

There was a diversity of opinion about whether the course got them interested or involved, perhaps because it was a required course for most of the graduate students. Nearly all agreed that the class required considerably more work than similar classes they have taken. It probably should be changed from a three-credit to a four-credit course.

Most of the students seemed to learn a lot, and could do fairly sophisticated problems by the end of the course. A few students never seemed to develop the facility, however. Part of this may have been my assumption of too much prior knowledge, so that theses students became overwhelmed and never caught up. In addition, most biology students are not used to the idea that they need to get things exactly right, otherwise the code won't work.

These difficulties may have been exacerbated by the fact that this has been taught as an on-line course, though under other circumstances it could certainly be taught in a regular classroom setting. Some students have indicated that they would like regular face-to-face sessions, but arranging the timing has proven difficult. A listserv makes asking and answering questions prompt and straightforward, and the students help each other both on the listserv and in group study sessions; but some of the explanatory and motivational things that typically go on in class may get shortchanged in the on-line format.

4 Conclusion

Overall, the course has been successful in teaching students from a variety of disciplines to use computational methods to model and analyze biological phenomena. R is an excellent computational tool for this purpose. The course covers a wide range of pertinent topics in biochemistry, molecular biology, and biophysics, many at a level thaat could not be adequately handled without computational tools. Other instructors could readily modify this list of topics to meet local needs and interests. Students who master this material are well-prepared to use high-level computational approaches to modern biology as their careers progress.

References

1. National Research Council: BIO 2010: Transforming Undergraduate Education for Future Research Biologists (2003)
2. The Comprehensive R Archive Network, `http://cran.r-project.org/`
3. The Bioconductor Project, `http://www.bioconductor.org/`
4. Bloomfield, V.A.: Computer Simulation and Data Analysis in Molecular Biology and Biophysics: An Introduction Using R. Springer, Heidelberg (2010) (in press)
5. `http://www.rseek.org/`
6. Search the R Statistical Language, `http://www.dangoldstein.com/search_r.html`
7. Girke, T.: R & Bioconductor Manual, `http://faculty.ucr.edu/tgirke/Documents/R_BioCond/R_BioCondManual.html`
8. Short, T.: R Reference Card, `http://cran.r-project.org/doc/contrib/Short-refcard.pdf`

9. Dalgaard, P.: Introductory Statistics with R. Springer, Heidelberg (2002)
10. Verzani, J.: Using R for Introductory Statistics. Chapman & Hall/CRC, Boca Raton (2005)
11. Deonier, R.C., Tavaré, S., Waterman, M.S.: Computational Genome Analysis: An Introduction. Springer, Heidelberg (2005)
12. Gentleman, R.: R Programming for Bioinformatics. Chapman & Hall/CRC, Boca Raton (2008)
13. Paradis, E.: Analysis of Phylogenetics and Evolution with R. Springer, Heidelberg (2006)
14. Gentleman, R., Carey, V.J., Huber, W., Irizarry, R.A., Dudoit, S. (eds.): Bioinformatics and Computational Biology Solutions Using R and Bioconductor. Springer, Heidelberg (2005)
15. Hahne, F., Huber, W., Gentleman, R., Falcon, S.: Bioconductor Case Studies (Use R). Springer, Heidelberg (2008)
16. Wilkinson, D.J.: Stochastic Modelling for Systems Biology. Chapman & Hall/CRC, Boca Raton (2006)
17. Keen, R.E., Spain, J.D.: Computer Simulation in Biology: A BASIC Introduction. Wiley-Liss, Chichester (1992)
18. Fall, C., Marland, E., Wagner, J., Tyson, J. (eds.): Computational Cell Biology. Springer, Heidelberg (2002)
19. Allman, E.S., Rhodes, J.A.: Mathematical Models in Biology: An Introduction. Cambridge University Press, Cambridge (2004)