EDUCATION

Computing Has Changed Biology—Biology Education Must Catch Up

Biologists need better computational education so that researchers can benefit from the bioinformatics revolution.

Pavel Pevzner[1]* and Ron Shamir[2]

Advances in computing have forever changed the practice of biological research. Computational biology, or bioinformatics, is as essential for biology in this century as molecular biology was in the last. In fact, it is difficult to imagine modern molecular biology without computational biology. For example, a difficult algorithmic puzzle had to be solved in order to successfully assemble the human genome sequence from millions of short pieces.

However, the computational components of undergraduate biology education have hardly changed in the past 50 years. New courses for biologists should be more relevant to their discipline, complementing the standard mathematical courses that were originally designed for physicists and engineers. Bioinformatics and biology communities should work together so that education of biologists in the 21st century may become as sophisticated as the computational education of physicists or economists.

For example, today's typical undergraduate economics curriculum may cover linear and integer programming, combinatorial algorithms, dynamic programming, game theory, and other computational concepts. These disciplines were in their infancy 40 years ago when the computational revolution started in economics. Because most biologists today (as most economists 40 years ago) do not know dynamic programming, for example, the idea of introducing such concepts into the biology curriculum may appear foreign and impractical. But the paradoxical result is that economics undergraduates may now be better prepared than biology graduate students to understand how DNA sequence alignment or gene prediction algorithms work (based on dynamic programming).

The RECOMB Bioinformatics Education Conference (http://casb.ucsd.edu/bioed/) explored ways to teach bioinformatics to undergraduate biology students. Attending biologists, computer scientists, and mathematicians from various branches of bioinformatics agreed that the time has come to shift the paradigm in biology education by adding new computational courses to standard curricula. This realization is not new: *BIO2010*, a National Research Council report (*1*), recommended substantial changes in the mathematics curricula for research-oriented biology undergraduates. Bialek and Botstein (*2*) and Pevzner (*3*) acknowledged the problem and outlined some creative approaches to its solution. However, the question of how best to deliver computational ideas to biologists remains.

Because bioinformatics is a computational science, courses should strive to present the ideas that drive an algorithm's design and to explain the crux of a statistical approach, rather than merely to recount the algorithms and statistical techniques. It is critical that bioinformatics is taught as a science that explains computational ideas and shows how they pertain to biological problems, rather than as a collection of cookbook-style recipes. A course must not be reduced simply to "Using Bioinformatics Tools," because a protocol-centric, how-to approach to teaching bioinformatics (without explaining computational ideas) is not unlike teaching how to take integrals in a calculus course without explaining what an integral is. For example, biologists sometimes use bioinformatics tools in the same way that an uninformed mathematician might use a polymerase chain reaction (PCR) kit, without knowing how PCR works and without any background in biology.

Many undergraduate bioinformatics programs at leading universities involve a grueling mixture of biological and computational courses that prepare students for follow-up bioinformatics courses and research. But many such courses aimed at bioinformatics undergraduates tend not to be ideally suited for biology students (undergraduate or graduate). This leads to a pedagogical challenge that, to the best of our knowledge, has not been resolved. How should the research and education community design a bioinformatics course that (i) assumes few computational prerequisites, (ii) assumes no knowledge of programming, and (iii) instills in students a meaningful understanding of computational ideas and ensures that they are able to apply them?

Consider the problem of analyzing gene expression data by principal component analysis (PCA), a powerful computational technique used by thousands of biologists. PCA is not typically covered in mathematics courses taken by biologists, so many may use PCA without understanding how it works or even what it does. A biologist who "blindly" uses PCA or other bioinformatics tools may misapply the method, miss important observations, misinterpret the results, and derive erroneous biological conclusions [see (*4*) for examples of misinterpretations of BLAST results].

Thus, we believe that undergraduate curricula should contain an additional course, "Algorithmic, Mathematical, and Statistical Concepts in Biology" to present the underlying ideas that drive computations in the field. This would not necessarily mean, for example, that biologists need an entire course in linear algebra to introduce eigenvalues, a fundamental aspect of PCA. At the RECOMB conference, Martin Vingron proposed ideas that can allow a simpler, elegant, and intuitive geometric interpretation of eigenvalues to address the problem of sorting a matrix so that similar rows are adjacent, a key problem in gene expression analysis [see (*5*)].

Some Biology departments have made progress toward introducing such courses. They focus on biological questions (e.g., "Did our ancestors interbreed with Neanderthals?" or "How do we distinguish between different forms of breast cancer and choose the appropriate chemotherapy?"), then follow with the computational ideas used to answer them. The best such courses are often designed by a team of faculty from different departments (e.g., Biology, Computer Science, and Mathematics). For example, Wingreen and Botstein (*6*) describe a course at Princeton that covers dynamic programming, clustering algorithms, Bayesian analysis, and other computational ideas relevant to original path-breaking papers in diverse areas of biology.

Or take, for example, the problem of selecting expression biomarkers that can be used to predict clinical outcomes of young breast cancer patients (*7*). The computa-

[1]Department of Computer Science, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA. [2]The Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv, 69978, Israel.

*Author for correspondence. E-mail: ppevzner@cs.ucsd.edu

Published by AAAS

tional ideas (hierarchical clustering, pattern classification, and feature selection) are introduced as part of the "story" to promote ease of understanding by students. Such examples may also instill in students the real-life impact of computational methods. This research, for example, led to the first cancer diagnostic chip to be approved by the U.S. Food and Drug Administration; it is currently used to determine which patients may benefit from additional chemotherapy.

The question of whether similar innovative courses can be implemented at the undergraduate level at many universities remains subject to debate (*8*, *9*). Nevertheless, such courses are pioneering steps toward developing a new computational biology curriculum.

We do not argue against the mathematical courses included in current undergraduate biology curricula. But we believe that these courses should be revised and extended. Many key computational ideas can be better communicated and absorbed by biology undergraduates with few prerequisites, in a way that will make the students excited about bioinformatics as a scientific discipline and more creative when they employ bioinformatics methods and ideas in the future. We feel that the best way to engage biology undergraduates in bioinformatics is to appeal to their innate intuition and common sense and to avoid mathematical formalism as much as possible. The proposed course may become a first step toward building the new computational curriculum for biologists.

References and Notes

1. National Research Council, *BIO2010, Transforming Undergraduate Education of Future Research Biologists* (National Academies Press, Washington, DC, 2003).
2. W. Bialek, D. Botstein, *Science* **303**, 788 (2004).
3. P. A. Pevzner, *Bioinformatics* **20**, 2159 (2004).
4. L. M. Iyer *et al.*, *Genome Biol.* 2(12), RESEARCH0051.1 (2001).
5. P. Grindrod *et al.*, *Math. Today* **44**, 80 (2008).
6. N. Wingreen, D. Botstein, *Nat. Rev. Mol. Cell Biol.* **7**, 829 (2006).
7. L. J. van 't Veer *et al.*, *Nature* **415**, 530 (2002).
8. L. J. Gross, *Cell Biol. Educ.* **3**, 85 (summer 2004).
9. R. Brent, *Cell Biol. Educ.* **3**, 88 (summer 2004).
10. We are grateful to all participants of RECOMB Bioinformatics Education conference (La Jolla, 14 and 15 March 2009) for many comments on various aspects of bioinformatics education. We are also grateful to V. Bafna, N. Bandeira, A. Tanay, and G. Tesler for many interesting discussions and suggestions. The conference was supported by the Howard Hughes Medical Institute Professors Program.

10.1126/science.1173876

EDUCATION

Mathematical Biology Education: Beyond Calculus

Training in developing algebraic models is often overlooked but can be valuable to biologists and mathematicians.

Raina Robeva[1]* and Reinhard Laubenbacher[2]

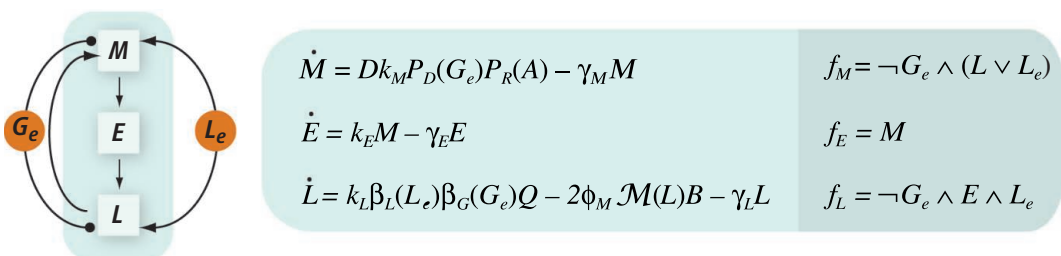
In 2003, the National Research Council's *BIO2010* report recommended aggressive curriculum restructuring to educate the "quantitative biologists" of the future (*1*). The number of undergraduate and graduate programs in mathematical and computational biology has since increased, and some institutions have added courses in mathematical biology related to biomedical research (*2*, *3*). The National Science Foundation (NSF) and the National Institutes of Health are funding development workshops and discussion forums for faculty (*4*, *5*), research-related experiences (*6*, *7*), and specialized research conferences in mathematical biology for students (*8*, *9*).

This new generation of biologists will routinely use mathematical models and computational approaches to frame hypotheses, design experiments, and analyze results. To accomplish this, a toolbox of diverse mathematical approaches will be needed.

Nowhere is this trend more evident than in systems biology. At the molecular level, this involves understanding a complex network of interacting molecular species that incorporates gene regulation, protein-protein interactions, and metabolism. Two types of models have been used successfully to organize insights of molecular biology and to capture network structure and dynamics: (i) discrete- and continuous-time models built from difference equations or differential equations (DE) models, which focus on the kinetics of biochemical reactions; and (ii) discrete-time algebraic models built from functions of finite-state variables (in particular Boolean networks), which focus on the logic of the network variables' interconnections.

Algebraic models were introduced in 1969 to study dynamic properties of gene regulatory networks (*10*). They have proven useful in cases where network dynamics are determined by the logic of interactions rather than finely tuned kinetics, which often are not known. Published algebraic models include the metabolic network in *Escherichia coli* (*11*) and the abscisic acid signaling pathway (*12*).

The use of algebraic methods is extending beyond systems biology. Methods from algebraic geometry have been used in evolutionary biology to develop new approaches to sequence alignment (*13*), and new modeling of viral capsid assembly has been developed using geometric constraint theory (*14*). Algorithms based on algebraic combinatorics have been used to study RNA secondary structures (*15*).



DE and Boolean models of the *lac operon* mechanism. Each component of the shaded part of the wiring diagram is a variable in the model, and the compartments outside of the shaded region are parameters. Directed links represent influences between the variables: A positive influence is indicated by an arrow; a negative influence is depicted by a circle.

[1]Department of Mathematical Sciences, Sweet Briar College, Sweet Briar, VA 24595, USA. [2]Virginia Bioinformatics Institute, Virginia Tech, Blacksburg, VA 24061, USA.

*To whom correspondence should be addressed. E-mail: Robeva@sbc.edu.