BIOINFORMATICS

Vol. 17 no. 9 2001
Pages 761–762

Editorial

TRAINING FOR BIOINFORMATICS AND COMPUTATIONAL BIOLOGY

The explosive increase in biological information produced by large-scale genome sequencing and gene/protein expression projects has created a demand that greatly exceeds supply for researchers trained both in biology and in computer science—two quite different disciplines. National funding agencies both in the US and Europe have committed significant resources to Bioinformatics training. Dozens of research universities have created new centers or programs in Bioinformatics, Computational Biology, and Genomics. Bioinformatics and Computational Biology training programs raise important questions: Whom should we train? What should they learn? Who should do the training?

Whom should we train? How we train students in Computational Biology and Bioinformatics depends, of course, on what we are preparing them to do, both as an intellectual discipline, and as a career path. Undergraduate or masters level training in Bioinformatics may have very different goals from doctoral or post-doctoral training in Computational Biology. The choice of undergraduate/masters vs. doctoral/post-doctoral training reflects another central question: Are Computational Biology and Bioinformatics tool-building/engineering disciplines? Or are they disciplines that address fundamental scientific questions and provide new insights into biological processes?

As someone who has spent the majority of my professional career doing research and publishing papers in Computational Biology, I feel strongly that Computational Biology research addresses fundamental biological problems. But there is clearly a substantial technical/tool-making dimension to most problems in Computational Biology—I am perhaps best known for developing a popular tool, the FASTA sequence comparison program. Identification of distantly related protein sequences, gene-finding, or protein structure prediction all demand better and better computational tools, not only to support research in other areas of structural and molecular biology, but also to understand the fundamental processes of protein evolution, protein folding, and gene transcription and splicing.

Indeed, the situation in Computational Biology today resembles that in Molecular Biology twenty years ago, when Assistant Professors were recruited in part because they understood a new technology (recombinant DNA) but also because they could use that technology to address traditional problems in Biochemistry, Genetics, and Molecular Biology. Now, as then, Computational Biologists who address challenging biological problems with innovative experimental approaches are much more likely to be successful at building independent research careers than computer-savvy biologists with tools looking for a problem.

However, there are substantial differences between the demand for ‘Cloners’ twenty years ago and for Computational Biologists and Bioinformatics researchers today. Recombinant DNA revolutionized Molecular Biology, but it did not immediately change the scale of biological research. Research groups moved from purifying to cloning their favorite proteins, but they studied the same proteins. Genome biology presents a different scale, whose promise will not be fulfilled without an infrastructure of well-trained researchers in Bioinformatics, Computational Biology, and Biomathematics, who are comfortable manipulating and analyzing large biological data sets. Moreover, most of the techniques of recombinant DNA technology—such as bacterial culture and transformation and DNA purification, separation and hybridization—were familiar to local faculty. Biochemists adopting recombinant DNA techniques may have needed to consult Microbiologists, but not the Mathematics Dept. Genome-scale Computational Biology is a much bigger change.

Thus, I believe the major goal for Computational Biology and Bioinformatics training should be to produce Ph.D. and post-doctoral fellows who can build independent research careers in Life Sciences Departments. The emphasis should be on research training, rather than mastering programming languages, analysis tools, and database administration.

What should they learn? If genome data are to be used effectively, what do Life Scientists need? Clearly, they need technicians to build World Wide Web interfaces, automate BLAST searches, and manage databases ranging from expression results to Laboratory Information Management Systems. These are routine tasks for someone with a Computer Science degree. But someone who is also comfortable with such biological concepts as sequence similarity and mRNA abundance can do them far more effectively.

From this perspective, an individual with several years of Computer Science training who has taken introductory courses in Biochemistry or Molecular Biology and Genetics, a course in Biological Sequence Analysis, and an introduction to Biostatistics can substantially enhance a laboratory’s or department’s ability to store, retrieve, and manipulate the results of genome-scale analyses. Indeed, individuals with this mixture of Computer Science and Molecular Biology training are in great demand. But I think it would be a mistake to build Bioinformatics programs to produce solely technicians who are not primary contributors to research question selection and

experimental design.

An alternative perspective, to which I subscribe, argues that Genomics, Computational Biology, and Bioinformatics offer fundamentally new opportunities for insights into Biological processes, which are difficult to identify and exploit without in-depth training in biological research.

A crucial difference between programs in experimental Biology and Computer Science is the relative amount of time devoted to hands-on research. Biological data are inherently variable, even in the most controlled experiments. Large amounts of biological data can magnify, rather than reduce, variation. Designing hypotheses and computational strategies that are robust to biological variation can be very difficult. Although programs in the Life Sciences differ, most graduate students in Biochemistry, Molecular Genetics, and Cell Biology spend 75–80% of their graduate career doing experiments at the lab bench; often less than a year is spent in the classroom. This heavy emphasis on practical experimentation reflects the experience that over the course of undergraduate, graduate, and post- doctoral training, many of the biological 'facts' presented to students in introductory courses will prove incorrect in some important detail.

Moreover, in biological research it is very difficult to teach robust experimental design. During their graduate careers, biologists are misled dozens of times when interpreting their results. Hence, the de-emphasis of course work in favor of experiments. Near the end of their graduate tenure, biologists should have developed both a healthy skepticism and an understanding of how to construct hypotheses and design experiments that will produce reliable conclusions.

To exploit genome data effectively, we must not only train Computer Scientists to understand biological concepts, but we must train Biologists, with their experimental experience and intuition, to develop computational strategies for large-scale analyses. Such training will require introductory courses in programming and program design, as well as data modeling, algorithms, machine-learning, and, again, statistics. (Many of the fundamental advances in sequence analysis, particularly at the genome scale, are based on statistical insights, yet neither Computer Science nor traditional Biochemistry, Molecular Genetics, Pharmacology, or Physiology curricula typically include statistics courses.) Ultimately, however, the most successful Computational Biology and Bioinformatics researchers will be those who identify significant and challenging biological problems that can be addressed by novel genome-scale experimental strategies.

Who should do the training? With the shortage of trained faculty in Computational Biology and Bioinformatics, few institutions will have the luxury of choosing between faculty in Biology and Computer Science Departments to lead new programs in Bioinformatics. Indeed, both types of Departments can make strong claims for institutional leadership. Often, Computer Scientists take the lead in developing Bioinformatics courses, in part, perhaps, because they feel more comfortable with the algorithms and computational approaches that these courses present.

I believe that Bioinformatics and Computational Biology training programs are better led from a Biological, rather than a Computer Science, perspective. Although Bioinformatics databases, web sites, and analysis pipelines present interesting engineering problems, the most interesting problems from a Computer Science perspective, which are often rooted in machine-learning or combinatorial algorithms, have not produced the tools that have the greatest impact for Biologists.

Thus, I would argue that Life Sciences researchers, with their understanding of the strengths and weaknesses of different experimental approaches to biological problems, should train Computational Biologists. That training will certainly include more course work than the traditional Life Sciences graduate program, and those courses will be found in Computer Science, Systems Engineering, and Statistics Departments as well as in the Life Sciences. But, as with traditional Life Science graduate programs, there will be a strong emphasis on experimental design, implementation, and analysis that is essential for a research career.

Training today, training tomorrow... Of course, these opinions reflect my perception of Bioinformatics and Computational Biology today, at the end of 2001. Today, fewer than two dozen institutions in the world have senior Life Science faculty who consider themselves Computational Biologists and are not Structural Biologists. Few Life Sciences graduate students, and even fewer undergraduates, take a course in Biological Sequence Analysis. This will clearly change dramatically over the next five years as new faculty are recruited and as more and more computational approaches follow the success of BLAST and ClustalW and move from the Computational Biologist's toolbox to common use by a large fraction of Biologists. As we understand better which computational approaches have been most productive, and why, we will understand better how to develop Computational Biology and Bioinformatics curricula for students at every level. In the future, sophisticated statistical, computational, and database methods may be as commonplace in Molecular Biology and Genetics as recombinant DNA is today. Today, however, Computational Biology and Bioinformatics are research disciplines, and training in the field must include a substantial experimental component.

William R. Pearson
Department of Biochemistry and Molecular Genetics
University of Virginia