BIOINFORMATICS

Vol. 14 no. 7 1998
Pages 549–550

Editorial

A Curriculum for Bioinformatics: The Time is Ripe

There seems to be wide agreement within both industry and academia that there are not enough scientists adequately trained in bioinformatics or computational biology. This sentiment stems principally from the difficulties in finding employees, graduate students and post-docs with appropriate skills for joining research and/or development teams in this field. The recent drain of academics into industry threatens to reduce our ability to provide the training needed to meet the demand of the job markets. An obvious question is 'What is the proper curriculum for bioinformatics professionals?' At first, the idea of defining a curriculum for bioinformatics may seem premature. The very definition of bioinformatics is still the matter of some debate. Although some interpret it narrowly as the information science techniques needed to support genome analysis, many have begun to use it synonymously with 'computational molecular biology' or even all of 'computational biology'. For this discussion of curriculum, bioinformatics addresses problems related to the storage, retrieval and analysis of information about biological structure, sequence and function.

There are currently two models for training. In the first model, post-doctoral fellows with core training in a technical field (such as computer science) or in a subdiscipline of biology receive speciality training in computational biology in order to become a 'computer scientist who specializes in biology' or a 'biologist who specializes in computer science'. While a valuable strategy, the post-doctoral model suffers because it is an expensive way (both in time and resources) to train individuals — learning the 'other' field is in many cases like going back to graduate school. In the second model, therefore, graduate students are trained primarily in bioinformatics or computational biology, without a preliminary training in one of the contributing disciplines. The curriculum for these students must provide them with a skill set that is long-lived and endures beyond the current fads of what is considered 'hot'. These students will not have traditional biological science or technical training to fall back on, and so it is critical that we provide the next generation with skills to solve industrial and academic problems that we cannot anticipate. I approach the problem of defining curriculum with a bias that bioinformatics is not simply a proper subset of biology or computer science, but has a growing and independent base of scientific tenets that requires specific training not appropriate for either biology or computer science alone. An appropriate academic curriculum for the field requires that we recognize the role of contributing disciplines, as well as the rapidly forming core literature in bioinformatics.

We must be careful not to define 'curriculum' narrowly as a list of required courses. Other elements of training include career-counseling, exposure to the culture of the discipline, and access to quality research projects for training. I therefore outline a proposal for an academic curriculum for training both MS and PhD level students. The idea of training undergraduates in bioinformatics is intriguing, and may well parallel the program outlined here. The chief distinction between MS and PhD training in the United States is usually that an MS graduate is a competent practitioner of a field, while a PhD is able to conceive, execute and report on independent novel work in the field. (This proposal is based on a combination of experiences training students at Stanford.)

What kind of basic background should be required of graduate students in bioinformatics? Students should ideally have undergraduate exposure to the natural sciences (physics, chemistry, biology) as well as quantitative technical disciplines (computer programming, applied mathematics, basic statistics). In the graduate training, the course requirements for competency in computational biology or bioinformatics can be divided into five areas: biology, computer science, statistics, ethics and core bioinformatics.

1. The training in biology is meant to ensure that the practitioner has a sense both of the basic theoretical constructs in biology, as well as a sense of how biological experimentation is done. In general, courses in molecular biology, genetics and cell biology are useful for the current problems in bioinformatics, although this could change as fields and interests change.
2. The key aspects of computer science for computational biology include programming courses, data structures/algorithms, databases and elective courses based on the interests of the student (robotics, numerical analysis, optimization, or artificial intelligence). The elements that are probably not required from standard computer science training might be the traditional operating systems and compiler courses that play a relatively smaller role in bioinformatics.
3. The statistics training should include courses in probability theory, experimental statistical design and analysis, and stochastic processes.
4. The ethics area should be designed to give students time to ponder the effects of technology on society, as well as covering issues of privacy and security.
5. The final and most critical portion of the curriculum would be a set of core bioinformatics courses that build upon the contributing disciplines to present the basic intellectual structure of the field. The precise list of core areas remains a topic of debate, but the recent release of a number of textbooks for bioinformatics (Baldi and Brunak, 1998; Bishop and Rawlings, 1997; Durbin *et al.*, 1998; Gusfield, 1997; Salzberg *et al.*, 1998; Setubal and Medianis, 1997) indicates that there is general agreement about the importance of certain fundamental concepts. These include (in no particular order):

- Pairwise sequence alignment (dynamic programming, heuristic methods, similarity matrices)
- Multiple sequence alignment
- Hidden Markov Models (construction, use in alignment, prediction)
- Phylogenetic Trees
- Fragment and map assembly and combinatorial approaches to sequencing
- RNA Secondary structure prediction
- Sequence feature extraction/annotation
- Protein homology modeling
- Protein threading
- Protein molecular dynamics
- Protein *ab initio* structure prediction
- Integration of molecular biology databases
- Support of laboratory biology (sequencing, structure determination, DNA arrays, etc.)
- Design and implementation of biological databases/knowledge bases.

In addition, the computer science and statistics sections of the curriculum should introduce the student to certain key technologies that are commonly used within bioinformatics, including:

- Optimization (Expectation Maximization, Monte Carlo, Simulated Annealing, gradient-based methods)
- Dynamic programming
- Bounded search algorithms
- Cluster analysis
- Classification
- Neural Networks
- Genetic Algorithms
- Bayesian Inference
- Stochastic Context Free Grammars.

Assuming that a typical course in a quarter system is 3 units, and that students take 9–12 units per quarter while working 6–9 units of research with a total load of 18 units per quarter, this curriculum could be finished in approximately two years (9 units of biology, 12 units of computer science, 9 units of statistics, 6 units of ethics and 9 units of core bioinformatics, along with 6–9 units of electives). At the end of this period, a comprehensive graduating exam for MS students or PhD candidates advancing to their dissertation research could be administered.

The next component of the curriculum is the introduction of the student to the culture of bioinformatics. First, journal clubs and advising sessions should expose the students to research problems in the field, and the ways in which they can be approached. Students should learn about the chief publications in the field, and the sources of research funding support. Second, the student should ideally attend one of the professional meetings in bioinformatics, in order to meet professional colleagues, and to get a sense for the dynamics of the discipline. Third, the student should be introduced to the opportunities and challenges in both academic and industrial settings as they prepare to make career decisions. Fourth, the student should have the opportunity to present both the work of others (in journal club settings) as well as their own work (in protected internal research colloquia) in order to develop the presentation skills that are clearly critical for success in both academic and industrial settings.

The final component of a bioinformatics curriculum is exposure of the student to original research projects. For MS students, this is a critical opportunity to get non-classroom training in the field, and to participate in a research project. Clearly, for PhD students this is the time during which the skills of independent investigation and scholarship are refined. A bioinformatics student may need more than one advisor for optimal training: the interdisciplinary nature of the field is often manifested in research projects that are jointly administered by biologists and computer scientists. The students need to have access to a broad variety of research projects and mentors, and must have the opportunity to excel in these projects, make independent contributions, and report their results in published and oral forms.

There are two other issues related to the creation of a curriculum in bioinformatics that should be addressed. First, we need clear career pathways for both academic and industrial scientists. A proper classification of these scientists in industrial research and development units should recognize their skills and contributions. Similarly, there should be a model for scientists to work within academic units where their work can be appropriately judged, with clear promotion criteria. Second, we need to identify stable funding sources for research in our field. Professional societies such as the ISCB must educate funding agencies about the importance of having a diverse portfolio of projects across all areas of biological science, including many technical approaches.

References

Baldi,P. and Brunak,S. (1998) *Bioinformatics: The Machine Learning Approach*. MIT Press, Cambridge, MA.

Bishop,M. and Rawlings,C. (eds) (1997) *DNA and Protein Sequence Analysis — A Practical Approach*. IRL Press at Oxford University Press.

Durbin,R. *et al.* (1998) *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge.

Gusfield,D. (1997) *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University Press, Cambridge.

Salzberg,S. *et al.* (eds) (1998) *Computational Methods in Molecular Biology*. Elsevier Science, New York.

Setubal,J. and Medianis,J (1997) *Introduction to Computational Molecular Biology*. PWS Publishing Company, Boston, MA.

Russ B. Altman
altman@smi.stanford.edu