

# COMPUTER SCIENCE AND BIOINFORMATICS

*By* JACQUES COHEN

---

*Computer  
scientists should  
be encouraged  
to learn biology  
and biologists  
computer science  
to prepare  
themselves for  
an intellectually  
stimulating and  
financially  
rewarding future  
in bioinformatics.*

---

IN BARELY HALF A CENTURY COMPUTER SCIENCE HAS GROWN FROM INFANCY TO MATURITY. EMPLOYMENT IN COMPUTER SCIENCE WAS ASSURED UNTIL A FEW YEARS AGO. TODAY, HOWEVER, LIKE IN THE 1960s, WHEN DEMAND FOR PHYSICISTS WANED, COMPUTER SCIENTISTS ARE EAGER TO EXPLORE NEW POSSIBILITIES IN PROMISING FIELDS. BIOLOGY AND ITS RELATED DISCIPLINES LIKE BIOINFORMATICS ARE AT THE TOP OF THE LIST. \_\_\_\_\_

ILLUSTRATION BY RICHARD DOWNS



Here, I explore a number of issues elicited by trying to determine how computing enhances biology and how biology enlivens computer science. How much effort would be expended in redirecting computer scientists to do work in bioinformatics? What bioinformatics topics are closest to computer science? Should computer science departments involve themselves in preparing their graduates for careers in this new field? And if so, what topics should they cover? Such questions reflect immediate concerns. Whether and how computer science research will find inspiration in biology is a long-term proposition. Thus, one must probe several related topics, including: the looming propagation of biology throughout the sciences; the cultural differences between computer science and molecular biology; the current goals of molecular biology; the Web data used in bioinformatics; the areas within computer science of interest to biologists; and the potential for employment in bioinformatics research, as well as in its commercial applications.

My interest in the subject stems from my years learning biology and teaching a course in computational biology for computer science majors at Brandeis University. More recently, I also co-taught courses in bioinformatics jointly offered by Brandeis and Wellesley College in which I established close working relationships with biologists. These relationships have given me insight into a different type of logical reasoning, unlike the one I was accustomed to in computer science.

Since the 1953 milestone achievement by James Watson and Francis Crick determining the structure of DNA, biology, especially molecular biology, has grown by leaps and bounds. The sequencing of the human genome represents one of its triumphs; the sequencing of dozens of other organisms has followed. Most of these successes would be unthinkable without computers, prompting several questions, including: What is the role of computers in biology? Is it like sending humans to the moon, namely a tool, yet only one among many? Developments in physics and mathematics preceded by centuries the genesis of modern computers; calculus and differential equations made tasks like space exploration considerably easier. There is no such previous work in biology. Molecular biology is by nature a science of the discrete, a property it shares with computing.

The rules of biology are generally deterministic,

though they also involve many exceptions. While DNA may be likened to a computer program, its activation engenders puzzling behavior; for example, a program generates parts of its own interpreter, as well as parallel processing, concurrency, semaphores, fault tolerance, and program regeneration.

Two eminent computer scientists, Donald Knuth of Stanford University and Leonard Adleman of the University of Southern California, have emphasized the importance of biology and its connectedness with computer science. Knuth anticipates that the number of radically new results in pure computer science is likely to decrease, while scientists will continue working on biological challenges for the next 500 years [8]. Adleman has argued that biological life can be equated with computation [1]. These views suggest that biological problems will significantly influence future directions in computer science research, including, for example, DNA computing, variants of the  $\pi$ -calculus, and amorphous computing.

Computer science includes two main categories—theoretical and experimental—both closely related to the concept of algorithms. In the first, computer science researchers wish to classify algorithms and develop new, more efficient ones; in the second, they aim to facilitate human-computer interaction by developing useful tools. Developments in the theoretical component are guided by mathematics, whereas designers and programmers producing software depend on sound engineering practices. The ultimate test for successful developments in theory is correctness and generality, usually assessed through mathematical techniques. In the experimental, acceptance by a large number of practitioners is one criterion for judging success; economics is another.

Acceptance and economics do not play parallel roles in the natural sciences; for example, the ultimate test in biology, physics, and chemistry is a validating laboratory experiment, with nature as supreme referee. In computer science, researchers and practitioners alike favor generality and abstraction; the approach is often top-down, as if they were developing a program. In contrast, biologists generally favor a bottom-up approach. This is understandable; the minutiae are so important, and biologists are often involved in time-consuming experiments that might yield ambiguous results. Synthesis must eventually take place, but biologists are wary of generalizations.

These differences are crucial in explaining why

---

Biologists are aware of the degree of difficulty—in days, months, or y

computer scientists' reasoning differs from biologists' reasoning. Although each group follows arguably logical steps, biologists are aware of the degree of difficulty—in days, months, or years—in validating a given conjecture by lab experiment. This perception requires years of experience to develop and simply cannot be gathered in one huge knowledge database. Therefore, to participate in the field of bioinformatics, computer scientists must interact routinely with biologists.

When one field blends with another, the usually hyphenated term reflecting the combination carries an ambiguous meaning; for example, does bio-physics belong to the corpus of knowledge of physics or of biology? Is it an independent new discipline? Probably with time and success, the new-discipline interpretation holds. I suspect mathematical biology requires a great deal more knowledge of mathematics than of biology. Similarly, computational biology is being developed by computer scientists to satisfy the needs of biologists but basically requires extensive knowledge of computer science theory [6].

Biologists are not particularly interested in computer science theory for solving their day-to-day problems. The term “bioinformatics” is more appealing to biologists, as the “computation” in “computational biology” is not quite on target. Incidentally, the suffix “informatics” is probably of European origin; the word “informatique” denotes computer science in French. The aims of bioinformatics, as conceived by biologists, are outlined in [10].

### Molecular Biology

Metabolic and signaling pathways can be viewed as flowcharts describing the behavior of cells and their interactions with the environment. A major goal in molecular biology is functional genomics, or the study of the relationships among genes in DNA and their function. Gene function can be viewed through several prisms. A common interpretation is that function describes the role of a gene product, usually a protein, in reacting with other proteins in a metabolic or signaling pathway. However, molecular biologists know that protein interactions are dependent on protein structure, or shape.

Functions can also be conveyed through annotations written by researchers who have studied in detail a given protein and its interactions with other proteins. The notion of function is essentially related to

protein shape and to the behavior of the organs that make up a living being; for example, the study of cell differentiation in original stem cells is related to cell function.

Biologists and computer scientists may conclude that the ultimate objective of functional genomics is: Given the DNA of an organism, produce a simulator for a cell of that organism. That simulator (or flowchart representing metabolic and signaling pathways) embodies all that it knows about a cell's behavior, allowing in-silico experiments that enable biologists to bypass costly and ethically sensitive in-vitro or in-vivo trials. We are far from this goal, but it is an area where computer science can provide considerable research impetus.

In defining the problems in bioinformatics and functional genomics, I first briefly describe the kind of data available throughout the Web. However, this data is often incomplete in the sense there is significantly more data available about DNA than there is about the structure of proteins and their interactions.

Biologists deal with essentially four types of data structures:

- Strings.* To represent DNA, RNA, and sequences of amino acids;
- Trees.* To represent the evolution of various organisms;
- Sets of 3D points and their linkages.* To represent protein structures; and
- Graphs.* To represent metabolic and signaling pathways.

Furthermore, biologists are often interested in substrings, subtrees, subsets of points and linkages, and subgraphs. Strings (such as words and phrases) are also used to express annotations that convey a meaning given by researchers, though such meanings are sometimes vague and incorrect. Biological data is often characterized by huge size, the presence of laboratory errors (noise), duplication, and sometimes unreliability.

For inferring function from the existing data, a biologist must consider three factors:

- Genes, or substrings of DNA capable of generating proteins;
- Protein structures represented in 3D space; and
- The roles of these proteins within metabolic and signaling pathways.

ears—IN VALIDATING A GIVEN CONJECTURE by lab experiment.

Since data about protein shape and pathways is often unavailable, the detective work in bioinformatics consists of deducing possible function from existing, albeit limited, information. Evolutionary data plays a critical role in that deduction. A typical example is a human gene for which there is no known protein structural data or pathways. However, corresponding data may be available for other organisms (such as the mouse and the worm). Inferring function from them might save biologists much tedious laboratory work. In addition, the inferred data might suggest key experiments that would help formulate a conjecture.

The use of microarrays to aid in inferring function from experiments is an important new development in molecular biology. Microarrays are silicon chips with tens of thousands of rows of tiny holes holding preprocessed material that can be activated by various types of samples. A scanner measures the degree of activation, and the data is downloaded onto a computer for subsequent analysis. Microarrays help biologists understand interactions among genes and are therefore instrumental in determining gene function and metabolic pathways. For example, the sample material spread onto such a microarray might originate from a cell with a deficient gene, an experiment that might yield valuable data about the effects of that deficiency. However, the data is notoriously noisy. Microarrays are of great interest to the pharmaceutical industry because they allow researchers to assess gene function in cells or organs subjected to a medication being tested. Moreover, drug companies are often interested in designing compounds that attach to proteins and help correct faulty gene behavior that causes disease.

### Algorithms in Bioinformatics

This discussion sheds light on algorithms that are of interest to biologists. The following are some of the most important algorithmic trends in bioinformatics:

- Finding similarities among strings (such as proteins of different organisms);
- Detecting certain patterns within strings (such as genes, introns, and  $\alpha$ -helices);
- Finding similarities among parts of spatial structures (such as motifs);
- Constructing trees (called phylogenetic trees)

expressing the evolution of organisms whose DNA or proteins are currently known;

- Classifying new data according to previously clustered sets of annotated data; and
- Reasoning about microarray data and the corresponding behavior of pathways.

The first three trends can be viewed as instances of pattern matching. However, pattern matching in biology differs markedly from its counterpart in computer science. DNA strings contain millions of symbols, and small local differences may be tolerated. The pattern itself may not be exactly known, because it may involve inserted, deleted, or replacement symbols. Regular expressions are useful for specifying a multitude of patterns and are ubiquitous in bioinformatics. However, what biologists really need is to be able to infer these regular expressions from typical sequences and establish the likelihood of the patterns being detected in new sequences.

This discussion suggests that both optimization and probabilistic approaches are necessary for developing biology-oriented pattern-matching algorithms. In the 1970s, a dynamic programming technique was devised to match two strings, taking into account the costs of insertions, deletions, and substitutions. Called global pairwise alignment, this technique was subsequently extended to consider local alignments, and, today, both methods are often used in bioinformatics [6]. However, dynamic programming is time consuming (it involves quadratic complexity) and therefore cannot be applied in a practical way to strings with hundreds of thousands of symbols.

A remarkable bioinformatics development from the 1990s is a pattern-matching approach called BLAST, or the Basic Local Alignment Search Tool, that mimics the behavior of the dynamic programming approach and efficiently yields good results. It is fair to say that BLAST is the most frequently used tool for searching sequences in genomic databases.

Another widely used and effective technique is multiple alignment, which helps align several sequences of symbols, so identical symbols are properly lined up vertically, with gaps allowed within symbols. The sequences may represent variants of the same proteins in various species; the goal is to find conserved parts of the proteins that are unchanged during evolution. Finding conserved parts of proteins also provides hints about a protein's possible function.

# Computer scientists are sure to benefit from being ACTI

Methods for multiple alignments are based on dynamic programming techniques developed for pairwise alignment.

After aligning multiple genomic or protein sequences, biologists usually depict trees representing the degree of similarity among the sequences being studied. Depicting evolutionary trees is in itself a domain within bioinformatics called phylogenetic trees. The problem of matching spatial structures can be viewed as a combination of computational geometry and computer graphics. Approximate methods are often required to find the longest linkage that is common in two 3D structures.

Bioinformatics involves the pervasive use of searches in genomic databases that often yield very large sets of long sequences. Such searches are often performed automatically by scripting to download massive amounts of genomic data from a number of Web sites. Script languages (such as Perl and Python) are often used for programming automatic searches in Web databases.

An approach commonly used in bioinformatics is: Given a human-annotated list of strings with boundaries specifying meaningful substrings—the learning set—now establish the corresponding likely boundaries for a new string; examples in bioinformatics involve finding genes and identifying the components of proteins.

Solutions to these problems are being explored through approaches from machine learning, neural networks, genetic algorithms, and clustering. Since the early 1990s a clustering technique called support vector machines (SVM) has had considerable success in biology. Classification and machine learning have been studied extensively in artificial intelligence to sort out new data based on a human-annotated set of examples.

Perhaps foremost among the machine learning techniques used in biology are the ubiquitous Hidden Markov Models, which are essentially probabilistic finite-state machines that use computed branching probabilities from a learning set and that establish the likelihood that a new string is processed through certain states with preestablished properties [4]. Hidden Markov Models were used in the 1980s for speech-recognition applications, thus demonstrating the serendipitous benefits of techniques that are transferable among different applications.

Two other algorithmic trends relevant to this dis-

cussion are related to microarrays and biologists' interest in computational linguistics. Recall that the main goal of analyzing microarray data is to establish relationships among gene behavior, possible protein interactions, and the effects of a cell's environment. From a computer science perspective, that goal amounts to the generation of parts of a program (flowchart) from data. This was also an early goal of program synthesis. However, it should be stressed that biological data is vast and noisy, spurring development of new techniques (such as Bayesian nets and SVM).

Information about the relationships among genes is often buried in countless articles describing the results of biological experiments. In the case of protein interaction, pharmaceutical companies have teams whose task is to search the available literature and “manually” detect phrases of interest. Efforts have been made to computerize these searches; their implementation requires expertise in both biology and computational linguistics.

### Answering the Original Questions

Computer scientists and biologists alike must realize that bioinformatics is not just a combination of using BLAST for pattern matching and Perl scripts for obtaining massive amounts of data from Web sites. It also takes sound knowledge of biology to make meaningful searches. In addition, the agent (whether human or program) invoking the search must be familiar with the limitations of the existing software packages and exercise judgment at each step taken in solving a problem.

*Employment in bioinformatics.* A computer scientist who is a specialist in systems, Web development, or computer graphics could quite possibly perform tasks in bioinformatics that do not differ significantly from those in other computer science applications. It is likely though, that additional knowledge of biology would enhance this person's employability potential in biotechnology. Having a programming background in script languages is a plus. However, programming per se without knowledge of the basics of biology limits significantly the interactions that are possible with coworkers.

Computer scientists wishing to specialize in the bioinformatics areas described here should have at least a background in biochemistry and take a course

# VE AND ASSERTIVE PARTNERS with biologists.



in bioinformatics, especially for working in the pharmaceutical industry. Knowledge of biochemistry is extremely helpful in understanding the details of biological structures, as well as of drug design. Additionally, knowledge of probability, statistics, and machine learning makes a candidate highly desirable for employment in bioinformatics. Redirecting a computer scientist to this endeavor can take approximately one year.

Several universities, including Stanford, Boston University, and the University of California, Los Angeles, now offer bioinformatics courses and majors. Short courses are also available at the main biological research laboratories, including Woods Hole, Cold Harbor, and Jackson. Self study is an option but lacks needed interaction with biologists. To start, I recommend perusing the *Cartoon Guide to Genetics* [5], followed by reading an undergraduate textbook (such as *Fundamental Concepts of Bioinformatics* [9]). The encyclopedic work by Mount [12] is useful to readers already familiar with biochemistry and who have had at least a course in molecular biology. A textbook by Jones and Pevzner [7] presents an algorithmic approach to the topic. My *ACM Computer Surveys* article [3] is designed to introduce bioinformatics to computer scientists who will find ample information about the current literature and textbooks, along with clues for the novice.

Learning bioinformatics can be arduous for computer scientists lacking formal training in biology; they would do best by having the attitude of trying to understand the reasoning and argot of biologists. They might be convinced—by proposing and implementing algorithmic solutions to biological problems—that computer science approaches can contribute significantly to producing fruitful results.

*Curricula.* The decision to teach bioinformatics within a computer science department is a function of the available resources. A department can take the following actions:

- Entice computer science majors to take a basic course in molecular biology and offer an introductory course in programming aimed at biologists;
- Convince computer science faculty to cover topics of interest to biologists, including machine learning, data mining, computer graphics, computational geometry, and Bayesian networks; and
- Introduce probabilistic and statistical approaches to algorithms and their data to cope with the often-imperfect situations that occur in biology.

## Conclusion

Bioinformatics is a developing interdisciplinary science. The involvement of other sciences (such as computer science) holds great promise; this century's major research and development efforts will likely be in the biological and health sciences. Computer science departments planning to diversify their offerings can thus only gain through early entry into bioinformatics. Even using minimal resources, such efforts are wise, as computer science graduates will enhance their employment qualifications.

Still unclear is whether bioinformatics will eventually become an integral part of computer science (in the same way as, say, computer graphics and databases) or will develop into an independent application. Regardless of the outcome, computer scientists are sure to benefit from being active and assertive partners with biologists. ■

## REFERENCES

1. Adleman, L. Computing with DNA. *Sci. Am.* 279, 2 (Aug. 1998), 54–61.
2. Cohen, J. Guidelines for establishing undergraduate bioinformatics courses. *J. Sci. Edu. Tech.* 12, 4 (Dec. 2003), 449–456.
3. Cohen, J. Bioinformatics: An introduction for computer scientists. *ACM Comput. Surv.* 36, 2 (June 2004), 122–158.
4. Durbin, R., Eddy, S., Krogh, A., and Mitchison G. *Biological Sequence Analysis*. Cambridge University Press, Cambridge, England, 1998.
5. Gonick, L. and Wheelis, M. *The Cartoon Guide to Genetics*. Harper Collins, New York, 1991.
6. Gusfield, D. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, Cambridge, England, 1997.
7. Jones, N. and Pevzner, P. *An Introduction to Bioinformatics Algorithms*. MIT Press, Cambridge, MA, 2004.
8. Knuth, D. Computer literacy interview (Dec. 7, 1993); [www.literateprogramming.com/clb93.pdf](http://www.literateprogramming.com/clb93.pdf).
9. Krane, D. and Raymer, M. *Fundamental Concepts of Bioinformatics*. Addison Wesley-Benjamin Cummings, Boston, 2003.
10. Luscombe, N., Greenbaum D., and Gerstein, M. What is bioinformatics? A proposed definition and overview of the field. *Methods Inf. Med.* 40, 4 (2001), 346–358; [bioinfo.mbb.yale.edu/papers/whatis-mim/](http://bioinfo.mbb.yale.edu/papers/whatis-mim/).
11. Mitchell, T. *Machine Learning*. WCB-McGraw-Hill, Boston, MA, 1997.
12. Mount, D. *Bioinformatics: Sequence and Genome Analysis, 2nd Ed.* Cold Spring Harbor Press, Cold Spring Harbor, NY, 2004.

---

JACQUES COHEN ([jc@cs.brandeis.edu](mailto:jc@cs.brandeis.edu)) is the TJX/Feldberg Professor of Computer Science in the Department of Computer Science at Brandeis University, Waltham, MA.

---



---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

---