The Emerging Landscape of Bioinformatics Software Systems

Fusing computing and biology expertise, bioinformatics software provides a powerful tool for organizing and mining the vast amounts of data genetics researchers are accumulating.

Lenwood S. Heath
Naren Ramakrishnan
Virginia Tech

Biology has increasingly become a data-driven science. The emergence of high-throughput data acquisition technologies for investigating biological phenomena has been an important factor in this process. For example, the recent completion of the human, rice, and flowering plant *Arabidopsis thaliana* genome maps signifies a major milestone in data acquisition.

The development of novel algorithms and databases to catalog, organize, harness, and mine the increasing amount of data such research efforts generate has also been important. The potential for scientists to infer significant biological knowledge computationally from a desktop is both appealing and real.

In this issue, we gather perspectives, articles, and reports to showcase the emergence of bioinformatics software as a discipline in its own right. The "Molecular Biology for Computer Scientists" sidebar provides some pertinent background information.

BIOINFORMATICS SOFTWARE SYSTEMS THEMES

As life scientists and computational scientists interact to create useful bioinformatics software systems, several themes or lessons recur. We identify seven themes:

- the nature of biological data;
- data storage, analysis, and retrieval;
- computational modeling and simulation;
- biologically meaningful information integration;
- data mining;
- image processing and visualization; and
- closing the loop.

Each of this issue's cover features touches on one or several of these themes.

The nature of biological data

The life sciences literature presents biological results based on carefully collected, analyzed, and vetted data that researchers can use with high confidence. Everyday bioinformatics, however, deals with raw data collected from recently completed experiments in the form of images, charts, or numbers and with sequence data collected from a wide variety of online databases. We should regard such data with some skepticism.

Any significant collection of raw experimental data includes experimental errors—systematic or random. Obtaining statistically meaningful data requires careful experimental design and replication of results. On the other hand, experiments are expensive in terms of professional labor, reagents, equipment, and time. As a consequence, biological data is always incomplete.

0018-9162/02/$17.00 © 2002 IEEE

Molecular Biology for Computer Scientists

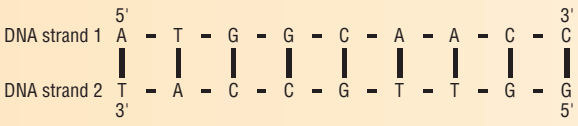
Every living organism can be properly classified into one of three *domains*[1]: Bacteria, Archaea, and Eucarya. A *eukaryote*—a member of the Eucarya domain—is a single- or multicellular organism, each cell of which has a bounding, semipermeable membrane and internal membrane-bound components, called *organelles*. The *nucleus* is one such organelle. It contains the organism's primary genetic information and the mechanism for genetic replication.

Plants and animals are eukaryotes. In contrast, every organism belonging to the Bacteria or Archaea domains is a *prokaryote*—a single-celled organism bounded by a semipermeable membrane that contains no internal organelles. In particular, a prokaryote has no nucleus. The first prokaryotic cells appeared on earth 3.5 billion years ago. Bacteria and Archaea evolved separately from a common ancestor approximately 3 billion years ago. Eucarya evolved from Archaea about 1.8 billion years ago.

Genomes and transcription

Chromosomes—large molecules of double-stranded deoxyribonucleic acid (DNA) and protein[2,3]—carry an organism's genetic information. Information on a DNA molecule takes the form of a linear code of four *bases*—A for adenine, C for cytosine, G for guanine, and T for thymine.

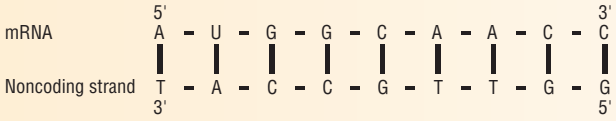
Pairs of *complementary bases* form weak intermolecular chemical bonds with each other. A complements T, while C complements G. In a chromosome, the two DNA strands each carry the same information, but one strand has the bases that complement the other strand's bases, in reverse order. For example, once reversed and complemented, the code ATGGCAACC becomes GGTTGCCAT, and the chemical bonding can be represented as follows:



DNA strands are oriented from a 5' end to a 3' end, so the codes on the two strands read ATGGCAACC and GGTTGCCAT.

An organism's *genome* consists of the entire DNA contents for all the chromosomes in any of its cells. A chromosome contains the organism's *genes*, which are particular subsequences of the linear code. Each gene includes *regulatory sequences* and an *open reading frame*. The ORF is the sequence within the gene that, in a process called *transcription*, is copied from the ORF on the *coding strand* of DNA to molecules called *messenger RNA* (mRNA).

Ribonucleic acid (RNA), a molecule similar to DNA, can also carry a linear code of four bases—A, C, G, and U for uracil. U replaces T as the base that complements A. Transcription occurs when an mRNA molecule is manufactured from the noncoding strand of DNA, as in this example:



The mRNA molecule therefore carries an exact copy of the ORF from the coding strand. *Post-transcriptional processing* may splice out some subsequences of the mRNA molecule, called *introns*. Eukaryotes and Archaea have introns, but Bacteria do not.

The protein factory

An mRNA molecule travels out of the nucleus into the *cytosol*, where the molecule's code determines how proteins are manufactured. The mRNA molecule's sequence is decoded to specify a sequence of 20 *amino acids* that form the protein's *primary structure*. A *codon*—a sequence of three bases drawn from A, C, G, and U—codes for an amino acid according to the *genetic code*.

For science to progress, we must combine inductive reasoning based on existing biological information with new experimental results. Indeed, some biological data is inherently unknowable, such as the genomes of most extinct species. Bioinformatics software system developers must always be aware that some uncertainty exists in any results the system generates. Thus, characterizing or quantifying the uncertainty is worth consideration.

Data storage, analysis, and retrieval

The high-volume, data-driven nature of modern experimental biology has led to the creation of many databases that contain genomes, protein sequences, gene expression data, and other data types. Researchers often key their retrieval of information from such databases primarily on one characteristic, such as the nucleotide or amino acid sequence, organism, gene annotation, or protein name.

Answering queries often involves some form of data analysis, such as statistical significance, clustering, or sequence homology search. The Basic Local Alignment Search Tool is typically the first bioinformatics tool a biologist uses when examining a new DNA or protein sequence. BLAST compares the new sequence to all sequences in the database to find the most similar known sequences.

Computational modeling and simulation

In addition to generating experimental data, computational simulation also plays a central role in understanding many biological processes. For example, researchers can study processes such as cell division by modeling reaction networks as a set of simultaneous differential equations.[1] They can then use tools from numerical and scientific computing to address questions such as "At what rate does this enzyme catalyze cell division?" The 1 March 2002 issue of *Science* showcases major developments in systems biology, many of which rely on simulation.

For example, AUG codes for the amino acid methionine (M), GCA codes for alanine (A), and ACC codes for threonine (T). Hence, the example mRNA fragment AUGGCAACC codes for the amino acid sequence MAT.

The *translation* process uses an mRNA molecule's genetic code to manufacture a protein molecule. Proteins are molecules essential to cell function. A protein may be an *enzyme* that catalyzes a biochemical reaction, a cell *structural component*, a *transducer* that responds to the environment, or a *regulatory element* that empowers or inhibits some process within the cell such as metabolism, transcription, or translation. Once manufactured, a protein folds into a three-dimensional shape whose geometric and chemical properties determine the protein's *function*.

Not all genes are active within a cell at any given time. In particular, *transcription factors*—molecules that differentially bind to the genes' regulatory sequences—control or *regulate* a gene's transcription. A gene that has been transcribed into mRNA is said to be *expressed*. Within a cell, hundreds or thousands of genes may be expressed at any time. In principle, to determine whether a gene is currently expressed and to what extent, researchers can measure how many mRNA molecules with the same coding sequence as the gene there are at any given time.

Evolving technologies

Two rapidly evolving technologies account for much of the burgeoning supply of basic biological data.[4]

First, *sequencing* can determine the sequence of bases in a DNA molecule that is a few hundred to a few thousand bases long. Researchers can cut a large DNA molecule such as a chromosome into overlapping fragments of manageable sizes, sequence all the fragments, and reconstruct the original molecule computationally. Sequencing and the associated computations are now so sophisticated that we can reconstruct a small genome a few million bases long in a few days.

Second, *microarrays* provide a method for identifying the expression of many or even all of an organism's genes in parallel at any given time.[5] Biologists can now obtain a snapshot or pattern of gene expression corresponding to the organism's response to a particular experimental condition. Unlike a genome, which provides only static sequence information, microarray experiments produce gene expression patterns that provide dynamic information about cell function. This information is essential to investigating complex interactions within the cell.

Through mutations in genomes, new species of organisms evolve. If we regard the first species of Earth's one-celled organisms as the root of a binary tree, all the species that currently exist or ever have existed form an *evolutionary* or *phylogenetic tree*. Because two species in close proximity on an evolutionary tree have genomes that are close in sequence, researchers can—and indeed must—study DNA and protein sequences to reconstruct phylogenetic relationships among species.

References

1. C.R. Woese, O. Kandler, and M.L. Wheelis, "Towards a Natural System of Organisms: Proposal for the Domains Archaea, Bacteria, and Eucarya," *Proc. Nat'l Academy of Sciences USA*, Nat'l Academy of Sciences, Washington, D.C., vol. 87, no. 12, 1990, pp. 4576-4579.
2. B. Alberts et al., *Essential Cell Biology*, Garland, New York, 1998.
3. G.M. Cooper, *The Cell: A Molecular Approach*, 2nd ed., ASM Press, Washington, DC, 2000.
4. G. Gibson and S.V. Muse, *A Primer of Genome Science*, Sinauer, Sunderland, Mass., 2002.
5. L.S. Heath et al., "Studying the Functional Genomics of Stress Responses in Loblolly Pine with the Expresso Microarray Experiment Management System," *Comparative and Functional Genomics*, vol. 3, no. 3, June 2002, pp. 226-243.

Biologically meaningful information integration

A tremendous quantity of valuable heterogeneous biological information can be found online, with bullish growth in the future a certainty. Examples of information sources include genomic sequences, gene sequences, expressed sequence tags, protein sequences, microarray experiment images and raw data sets, 2D protein gels, protein domains, and the literature of genetics, biochemistry, and molecular biology.

Researchers cannot answer the cutting-edge questions in biology with information from just two or three of these sources. A global database that integrates all these sources for all purposes is a pipe dream, given that we cannot predict in advance the myriad needs of biologists for accessing the information. However, resources dedicated to restricted domains—for example, BioCarta (http://www.biocarta.com/), an encyclopedia of regulatory pathways, and CyanoBase (http://www.kazusa.or.jp/cyano), a Web resource for cyanobacterial research—have recently become available. Thus, bioinformatics software systems increasingly face the task of integrating information from diverse sources. Further, their developers must address the challenge of obtaining biologically meaningful and useful results from those many sources.

Data mining

Bioinformatics systems benefit from the use of data mining strategies to locate interesting and pertinent relationships within massive information. For example, data mining methods can ascertain and summarize the set of genes responding to a certain level of stress in an organism. Researchers can use graphical models such as Bayesian networks and relational algorithms such as inductive logic programming to mine such gene sets and model a gene expression network. Even a cursory glance through

Bioinformatics software must support the strongly iterative and interactive nature of biology research.

the literature in journals such as *Bioinformatics* reveals the persistent role of data mining in experimental biology. Integrating data mining within the context of experimental investigations is central to bioinformatics software.

Image processing and visualization

Many results in experimental biology first appear in image form—a photo of an organism, cells, gels, or microarray scans. As the quantity of these results accelerates, automatic extraction of features and meaning from experimental images becomes critical.

At the other end of the data pipeline, naive 2D or 3D visualizations alone are inadequate for exploring bioinformatic data. Biologists need a visual environment that facilitates exploring high-dimensional data dependent on many parameters.

Closing the loop

Bioinformatics software must support the strongly iterative and interactive nature of biology research. Biologists typically revise and redesign experiments based on results from previous experiments. Providing feedback to earlier stages of an experiment based on downstream data—for example, to reorganize a microarray layout or alter dye concentrations—is central to improving the efficiency of biological investigation.

IN THIS ISSUE

The best bioinformatics software systems address the problems in bioinformatics within a context of insights into the

- computational complexities of those problems, and
- sophisticated knowledge of biology and current experimental technologies.

In this spirit, Mihai Pop, Steven L. Salzberg, and Martin Shumway present a tantalizing view of the development of sequence assembly systems for entire genomes. These systems must recognize the strengths and drawbacks of experimental technology and the challenging nature of real genomes, such as the existence of long tandem repeats.

In "Genome Sequence Assembly: Algorithmic Issues and Practical Considerations," the authors describe the techniques brought to bear on the computational issues of sequence assembly, including the theory of computation as it applies to NP-hardness, graph theory, and combinatorial algorithms; quality-assessment statistics; and heuristics that support tradeoffs among genome coverage, the number of misassemblies, and the number of contigs—a group of overlapping regions of a genome. Sequence assembly is typical of most bioinformatics problems in that the correct answer is perhaps either unknowable or only obtainable by paying the high cost of manual intervention.

The Assembling the Tree of Life (ATOL) project challenges biologists and computer scientists to use sequence and other biological information to determine the evolutionary relationships among existing species. ATOL further challenges researchers to represent these relationships in a tremendous evolutionary, or phylogenetic, tree. In "Toward New Software for Computational Phylogenetics," Bernard M.E. Moret, Li-San Wang, and Tandy Warnow present significant insights into the nature of ATOL's computational challenges and describe their progress in addressing those challenges.

Algorithms for reconstructing phylogenetic trees from sequences must scale to very large sequence sets. Moret and colleagues have thus developed disk-covering algorithms as a means for accomplishing scaling. High-performance algorithm engineering uses algorithmic and implementation savvy to produce highly efficient applications for challenging computational problems. The authors' Genome Rearrangement Analysis using Parsimony and other Phylogenetic Algorithms (GRAPPA) is a system that provides an excellent example of high-performance engineering and can teach valuable lessons to bioinformaticians.

In "BioSig: An Imaging Bioinformatic System for Studying Phenomics," Bahram Parvin and colleagues describe a system for archiving and interpreting microscopic images of small groups of cells drawn from mice and treated with ionizing radiation. Part of interpreting such an image involves segregating organelles, such as the nuclei, from the remainder of the image. Even smaller features, such as the chromatin—the chromosomes and associated proteins present in the nucleus when the cell is not reproducing—should also be identified as distinct from noise. The BioSig system for cell phenomics—the visual characteristics of a control or treated cell—can ultimately lead to predicting the effects of ionizing radiation in other mouse—and human—organs.

Meeting long-term bioinformatics goals requires reconciling information collected from studying biological phenomena at multiple scales and using multiple modes of investigation. For example, researchers can study biological processes at the

DNA, mRNA, protein, enzyme, pathway, reaction network, or physiology levels. Each level gives a different view to the underlying mechanism, but together they help establish the basis for answering biological questions.

In "A Random Walk Down the Genomes: DNA Evolution in Valis," Bud Mishra and colleagues present the Valis system, which prototypes bioinformatics applications, and describe its use for studying cellular events in relation to DNA sequence evolution. This article, which comes closest to our closing-the-loop theme, also describes a sophisticated modeling of sequence evolution. The authors cover the software design in detail and also describe their system's modeling and computational capabilities. Valis-like systems have the potential to model large-scale genomic processes, a Grand Challenge for bioinformatics.

BioSig and Valis constitute integrated problem-solving environments[2] for bioinformatics applications. These software systems provide all the computational facilities necessary for solving a target class of problems.

From performing functional studies on genes or gene families in isolation, research has progressed to studying all the genes in a given organism simultaneously. Microarray bioinformatics has aided in this massive parallelization of experimental biology.

In "Interactively Exploring Hierarchical Clustering Results," Jinwook Seo and Bernard Shneiderman present an interactive visualization system for investigating data from microarray experiments. The authors introduce the basics of microarray technology for nonspecialists and describe how to achieve user-driven interactive data exploration in relation to a hierarchical clustering algorithm. Interactive exploration and visualization will become increasingly important as the dimensionality and diversity of the underlying data increase.

In addition to these theme articles, this issue of *Computer* also features two perspective articles on bioinformatics. In "Computers Are from Mars, Organisms Are from Venus," Junhyong Kim examines the new relationships between biology and computer science. Kim identifies benefits that each field can draw from the other, as well as the challenges in interdisciplinary research. The essay is an excellent starting point for new researchers, and it provides the necessary background to appreciate the five theme articles.

In "The Blueprint for Life?" Dror G. Feitelson and Millet Treinin challenge the assertion that DNA encodes everything needed to understand life. The authors examine how information is interpreted, transported, and communicated in biological systems, and conclude that there is more about these processes than is encoded in DNA.

We received 20 submissions for this special issue, which we forwarded to reviewers from both biology and computer science backgrounds. We thank these reviewers for their timely responses and the authors for responding quickly to the reviewers' comments and suggestions.

Because this issue does not aim to provide comprehensive coverage of bioinformatics software, it does not address some important biological problems such as pathway modeling.[3] Likewise, it excludes several novel computational techniques, especially from data mining. The content does, however, reveal a glimpse into the richness of bioinformatics software, providing a snapshot of its continuing evolution. ■

From performing functional studies on genes or gene families in isolation, research has progressed to studying all the genes in a given organism simultaneously.

References

1. J.J. Tyson, K. Chen, and B. Novak, "Network Dynamics and Cell Physiology," *Nature Reviews Molecular Cell Biology*, Dec. 2001, pp. 908-916.
2. E. Gallopoulos, E.N. Houstis, and J.R. Rice, "Computer as Thinker/Doer: Problem-Solving Environments for Computational Science," *IEEE Computational Science and Engineering*, vol. 1, no. 2, 1994, pp. 11-23.
3. P.D. Karp, "Pathway Databases: A Case Study in Computational Symbolic Theories," *Science*, vol. 293, 2001, pp. 2040-2044.

Lenwood S. Heath *is an associate professor of computer science at Virginia Tech. His research interests include algorithms, theoretical computer science, graph theory, bioinformatics, computational biology, and symbolic computation. Heath received a PhD in computer science from the University of North Carolina at Chapel Hill. He is a member of the IEEE, the ACM, SIGACT, and SIAM. Contact him at heath@cs.vt.edu.*

Naren Ramakrishnan *is an assistant professor of computer science at Virginia Tech. His research interests include problem-solving environments, mining scientific data, and personalization. Ramakrishnan received a PhD in computer sciences from Purdue University. He is a member of the IEEE Computer Society, the ACM, and the AAAI. Contact him at naren@cs.vt.edu.*