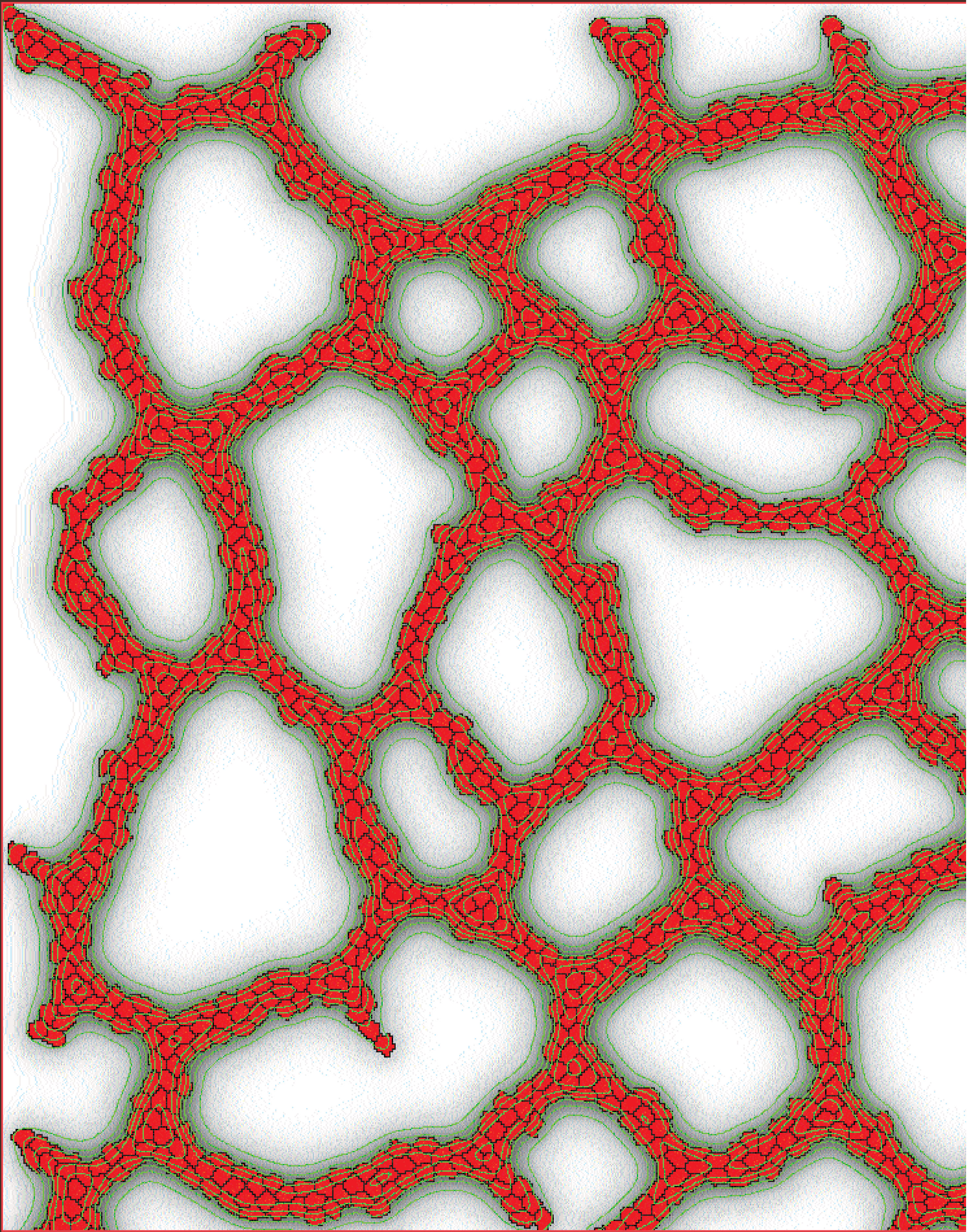
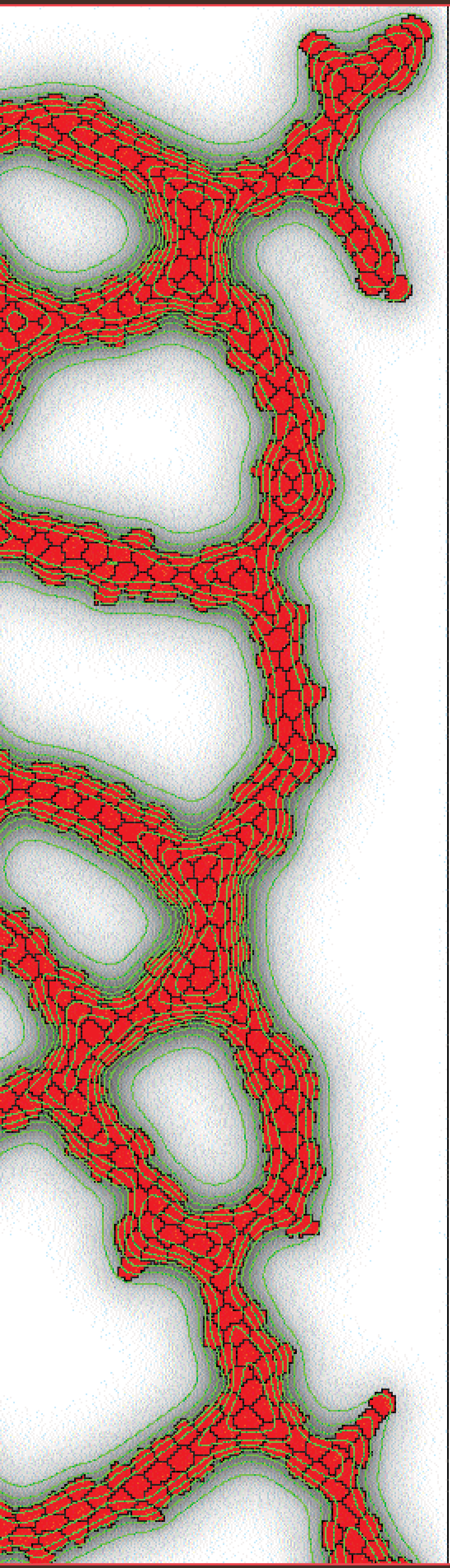




By David A. Bader

Computational Biology and High-Performance Computing

Understanding evolution and the basic structure and function of proteins are two grand challenge problems in biology that can be solved only through the use of high-performance computing.

Computational biology has been revolutionized by advances in both computer hardware and software algorithms. Examples include assembling the human genome and using gene-expression chips to determine which genes are active in a cell [11, 12]. High-throughput techniques for DNA sequencing and analysis of gene expression have led to exponential growth in the amount of publicly available genomic data. For example, the genetic sequence information in the National Center for Biotechnology Information's GenBank database has nearly doubled in size each year for the past decade, with more than 37 million sequence records as of August 2004. Biologists are keen to analyze and understand this data, since genetic sequences determine biological structure, and thus the function, of proteins. Understanding the function of biologically active molecules leads to understanding biochemical pathways and disease-prevention strategies and cures, along with the mechanisms of life itself.

Increased availability of genomic data is not incremental. The amount is now so great that traditional database approaches are no longer sufficient for rapidly performing life science queries involving the fusion of data types. Computing systems are now so powerful it is

Simulation of blood vessel formation through aggregation of dispersed endothelial cells. The model employs only experimentally confirmed behaviors of individual cells.
Roeland M.H. Merks and James A. Glazier, Biocomplexity Institute, Department of Physics, Indiana University, Bloomington.

possible for researchers to consider modeling the folding of a protein or even the simulation of an entire human body. As a result, computer scientists and biomedical researchers face the challenge of transforming data into models and simulations that will enable scientists for the first time to gain a profound understanding of the deepest biological functions.

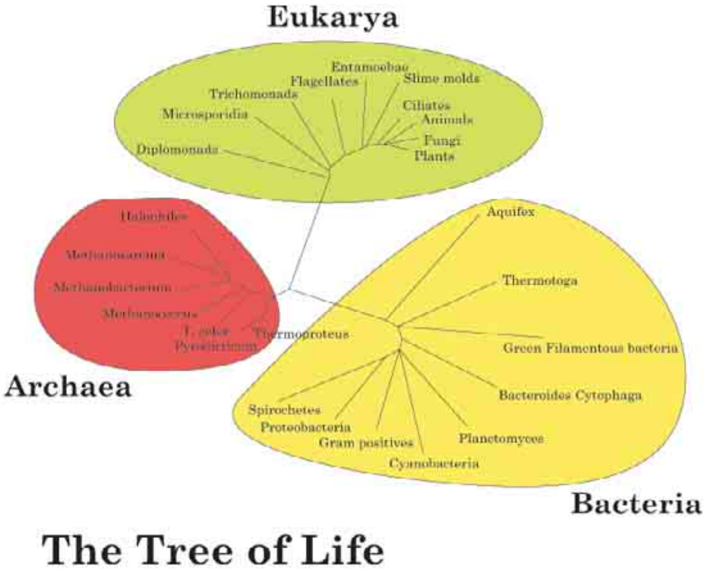


Figure 1. Tree of life, including: Eukarya, organisms with cells containing membrane-bound nuclei (such as animals, plants, fungi, and protists); Archaea, single-celled organisms that inhabit some of the most extreme environments on the planet; and Bacteria, single-celled organisms that are among the earliest forms of life, appearing on Earth billions of years ago. The tree represents the evolutionary relationships among all forms of life.

Traditional uses of high-performance computing (HPC) systems in physics, engineering, and weather forecasting involve problems that often have well-defined and regular structures. In contrast, many problems in biology are irregular in structure, are significantly more challenging for software engineers to parallelize, and often involve integer-based abstract data structures. Solving biological problems may require HPC due either to the massive parallel computation required to solve a particular problem or to algorithmic complexity that may range from difficult to intractable.

Many problems involve seemingly well-behaved polynomial time algorithms (such as all-to-all comparisons) but have massive computational requirements due to the large data sets that must be analyzed. For example, the assembly of the human genome in 2001 from the many short segments of sequence data produced by sequence robots required approximately 10,000 CPU hours [12].

Other problems are compute-intensive due to their inherent algorithmic complexity (such as protein folding and reconstructing evolutionary histories from molecular data). Some are known to be NP-hard (or harder). (An NP-hard problem is one for which an exact solution is conjectured by computer scientists to not be solvable in polynomial time, that is, an NP-hard problem requires more steps than can be grounded by a polynomial.) Thus, while NP-hard problems are thought to be intractable, HPC may provide sufficient capability for evaluating biomolecular hypotheses or solving more limited but meaningful instances.

Here, I investigate problems requiring massive parallelism due to their inherent algorithmic complexity (such as protein folding) or due to being NP-hard (such as inferring evolutionary histories from genetic information).

Protein Folding

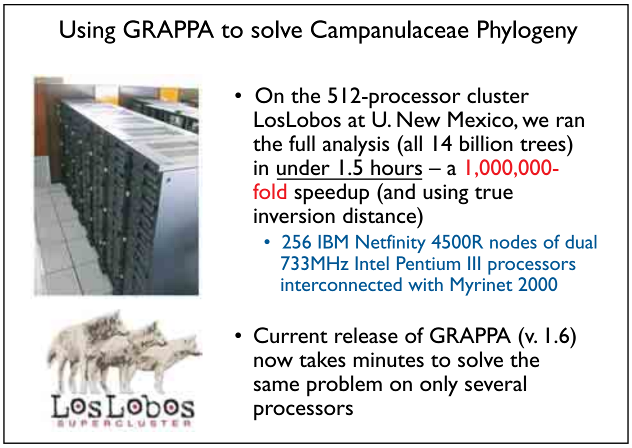
Proteins are large molecules found in all organisms built from a chain of amino acids and are responsible for the structure, function, and regulation of cells, tissues, and organs. Protein folding is the process of self-assembly of an amino acid sequence into the native 3D structure of the functioning protein. Proper functioning of a protein depends on its ability to fold into its native structure. Failure to do so causes a loss of biological function and often results in illness or fatal disease. Examples are cystic fibrosis; Parkinson's; Alzheimer's; and Prion diseases (such as Creutzfeldt-Jakob Disease and Bovine Spongiform Encephalopathy, or mad cow disease). Hence, a biomedical researcher's understanding of how a protein folds has direct medical significance. The protein-folding problem is computationally challenging, and many techniques, ranging from experimental to theoretical, are being investigated for their accuracy and speed in predicting 3D structures.

The process of folding a protein takes from approximately 20 microseconds to as much as one full second. While some proteins can be studied through X-ray crystallography, others (such as membrane proteins and molten globules where the side-chain packing in the interior of the fold is not in a rigid conformation) can be studied only through simulation [2]. Algorithms for protein folding span a broad range of sophistication (and computational cost) for

Because *phylogenies are crucial to answering fundamental open questions* IN BIOMOLECULAR EVOLUTION, *biologists have a strong interest in algorithms that resolve ancient relationships.*

biomolecular modeling of the physical processes. At least three different approaches are being taken to develop innovative parallel computing systems to run these algorithms: the massive computational system called Blue Gene/L being developed by IBM; specialized hardware specifically for molecular dynamics (such as the MD-GRAPE, the Molecular Dynamics GRAvity Pipe, or PetaFLOPS special-purpose computer system being developed by IBM's Research Division and by the Institute of Chemical Research, or RIKEN, in Japan); and cycle-scavenging approaches exemplified by the Folding@Home project at Stanford University.

IBM's Blue Gene project (see the article by Toshikazu Ebisuzaki et al. in this section) focuses on building massively parallel PetaFLOPS-class supercomputers designed to perform protein-folding simulations, as well as model other biomolecular phenomena. These simulations employ a molecular dynamics approach to protein folding, starting with a model of an unfolded amino acid chain and the solvent molecules surrounding the chain in a cell. The forces on every atom in the amino acid chain and the solvent molecules around it are calculated through an approach called explicit-solvent. The expected movements of the atoms over each individual time step are calculated from these forces, a process that must be repeated many times.


LosLobos
SUPERCLUSTER

Using GRAPPA to solve Campanulaceae Phylogeny

- On the 512-processor cluster LosLobos at U. New Mexico, we ran the full analysis (all 14 billion trees) in under 1.5 hours – a 1,000,000-fold speedup (and using true inversion distance)
  - 256 IBM Netfinity 4500R nodes of dual 733MHz Intel Pentium III processors interconnected with Myrinet 2000
- Current release of GRAPPA (v. 1.6) now takes minutes to solve the same problem on only several processors

Figure 2. Using GRAPPA, or Genome Rearrangement Analysis through Parsimony and other Phylogenetic Algorithms, to reconstruct the phylogeny of Campanulaceae.

The IBM Blue Gene project [6] estimates that simulating 100 microseconds of protein folding takes 10^{25} machine instructions. This computation would take three years on a PetaFLOPS system or keep a 3.2GHz microprocessor busy for the next million centuries.

The MD-GRAPE project (see the article by Toshikazu Ebisuzaki et al. in this section) also takes a molecular dynamics approach to simulating protein folding. Unlike the Blue Gene family, the GRAPE and MD-GRAPE systems are capable of performing calculations only for dynamical systems. Efforts are under way in Japan to create a PetaFLOPS computing resource comprised of many MD-GRAPE boards to perform large-scale simulations of protein folding.

The Folding@Home project (folding.stanford.edu) takes a different approach, both in terms of assembling computer power and to the algorithms it employs. Folding@Home invites volunteer Internet users to download a screensaver program that receives parcels of work from the project server whenever the client computer is idle. When the computer finishes processing its parcel, it returns the result to the server and receives a new assignment. In this way, thousands of processors around the world together contribute to the process of simulating protein folding.

The Folding@Home project uses an ensemble dynamics method and does not model the solvent molecules explicitly. Its algorithm instead models the ways an amino acid chain may move based on calculation of free energy barriers that constrain the way proteins may fold. (Switching from one conformation to another is not possible due to the amount of energy required to make the transition and is referred to as the free energy barrier.) The majority of compute time is spent exploring the free energy wells and waiting for thermal fluctuations that bring the system across the barriers. The Folding@Home system has attracted thousands of volunteers to donate compute time; from October 2000 to September 2004, more than a million CPUs worldwide were contributed to the project, amounting to tens of thousands of CPU cycles each year, breaking the 100TFLOPS barrier on September 9, 2004. This cycle-donation approach is also used in several other branches of computational

SIMULATING 100 MICROSECONDS OF PROTEIN FOLDING COULD TAKE 10^{25} MACHINE INSTRUCTIONS, A COMPUTATION THAT WOULD TAKE THREE YEARS ON EVEN A PETAFLOPS SYSTEM OR *keep a 3.2GHz microprocessor busy for the next million centuries.*

biology, including FightAids@Home (fightaidsathome.scripps.edu/) in a search for chemical compounds that might interact with the HIV virus to treat AIDS.

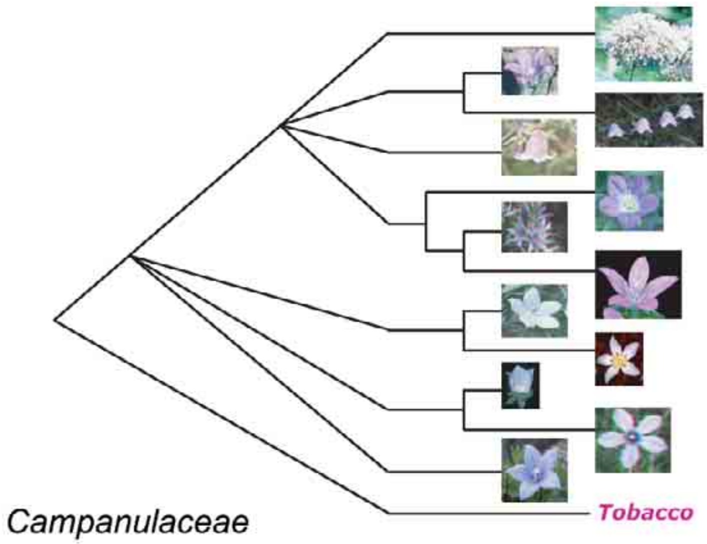


Figure 3. Bellflower family (Campanulaceae) and its many blooms. The evolutionary history reconstructed by GRAPPA depicted here confirms the conjecture that inversion is the principal process of genome evolution in cpDNA for this group.

IMAGE COURTESY OF BOB JANSEN, UNIVERSITY OF TEXAS AT AUSTEN, AND LINDA RAUBESON, CENTRAL WASHINGON UNIVERSITY

Phylogeny Reconstruction

A phylogeny is a representation of the evolutionary history of a group of genes, gene products, or species of organisms (taxa). Phylogenies have been reconstructed by biologists and paleontologists (without computing) for more than a century. The availability of genetic sequence data now makes it possible to infer phylogenetic trees from genetic sequences, or the sequences of molecules acted on directly by evolution and thus record evolution's end effects.

Phylogenetic analysis uses data from living organisms to attempt to reconstruct the evolutionary history of genes, gene products, and taxa. (Since the techniques are largely the same for all, I refer simply to taxa here). Because phylogenies are crucial to answering fundamental open questions in biomolecular evolution, biologists have a strong interest in algorithms that resolve ancient relationships. Much applied research depends on these algorithms as well. For example, pharmaceutical companies use phylogenetic analysis in drug discovery in, say, discovering biochemical pathways unique to target organisms. Health organizations study the phylogenies of organisms (such as HIV) to understand their epidemiologies and aid in predicting the course of disease over time in an individual or even in an entire population. Using an understanding of the phylogenetic distribution of variation in wild populations, government laboratories worldwide are working to develop improved strains of basic foodstuffs (such as rice, wheat, and potatoes). Finally, the reconstruction of large phylogenies has also yielded fundamental new insights into the process of evolution.

The basic principle of phylogenetic inference is that comparing genetic sequences makes it possible to find out which taxa are more closely related (and thus more recently separated in evolutionary time) and which are less closely related (and thus separated much further back in evolutionary time). Existing phylogenetic reconstruction techniques suffer from serious shortcomings of running time and accuracy, particularly for large data sets. Phylogenetic inference will benefit from new algorithmic developments, as well as from HPC systems that reduce the running time of current algorithms.

Optimization Problems

Almost every model of speciation and genomic evolution used in phylogenetic reconstruction has given rise to NP-hard optimization problems. Three major classes of methods are used by computer scientists in designing algorithms to solve them: heuristics (such as neighbor-joining) [9]; maximum parsimony [3]; and maximum likelihood [4]. Heuristics (a natural consequence of the NP-hardness of the problems) run quickly, but may not offer quality guarantees and even lack a well-defined optimization criterion. Parsimony-based methods take exponential time (as a function of the number of taxa), but, at least for DNA and amino acid, data is often run to completion on data sets of moderate size. Maximum-likelihood methods come with a larger set of conditions and assumptions than parsimony methods, but when these conditions are met, they seem to be capable of outperforming the others in terms of the quality of the solutions they produce. However, maximum-likelihood methods may take thousands of CPU hours to analyze large data sets (see the article by Mark Ellisman et al. in this section).

Until recently, most phylogenetic algorithms focused on DNA or protein sequence data using a model of evolution based mainly on nucleotide substitution. Another type of data has recently become available through the characterization of entire genomes: gene content and gene order data. For a few animal species (such as human, mouse, and fruit fly), several plants and microorganisms, and a fair sampling of cell organelles (mitochondria and chloroplast), biomedical researchers now have a thorough catalog of genes and their physical locations on chromosomes. Several mechanisms of evolution operate at the genome level, including gene duplication, loss, and reordering. They operate on time scales much slower than nucleotide mutations; as a result, they are potentially useful in resolving ancient evolutionary relationships. This new source of data has thus been embraced by biologists interested in the evolution of major divisions of plants, animals, and microorganisms.

Exploiting data about gene content and gene order has proved extremely challenging from a computational perspective. Tasks readily carried out in linear time for DNA data might require entirely new theories (such as computation of inversion distance [1, 5]) or appear to be NP-hard. Thus gene-ordering approaches have been used most extensively on simple genomes: bacteria and organelles (chloroplast and mitochondria). Mitochondria are organelles found in all eukaryotic cells in plants, animals, and protozoans and are instrumental in processing energy in cells (see Figure 1). Chloroplasts are found in photosynthetic protozoans and plants and are responsible for turning sunlight into energy-storing compounds. Mitochondria and chloroplasts have their own bacteria-like DNA. The genetic information of bacteria, mitochondria, and chloroplasts consists of a single chromosome and, unlike eukaryotes, all of their genetic material is expressed, meaning that all of the genomic coded information is transcribed and converted into the structures present and operating in the cell. Moreover, the order of genes in prokaryote-like DNA (in bacteria, mitochondria, and chloroplasts) is especially important in gene expression and cell function.

The evolutionary relationships of bacteria may be studied by examining the order of genes in bacterial DNA. Since mitochondria and chloroplasts have their own DNA (independent of the genetic material of the eukaryotic organisms of which they are a part), the evolutionary relationships of the eukaryotes can likewise be studied by examining the gene order of their mitochondria or chloroplasts.

Gene order along a chromosome can be viewed as an ordering of signed integers, with each integer representing a single gene; the sign denotes the relative orientation of the gene along the DNA. A method called breakpoint phylogeny [10] infers the structure of phylogenetic trees based on analysis of changes in gene order.

GRAPPA, or Genome Rearrangement Analysis through Parsimony and other Phylogenetic Algorithms, I developed, along with a group at the University of New Mexico and the University of Texas at Austin, beginning in 2000, extends and makes more realistic the underlying evolutionary theory of breakpoint analysis and provides a highly optimized parallel program that performs well on a variety of supercomputer systems (see Figure 2) [7]. We have used GRAPPA on the University of New Mexico's 512-processor Linux cluster to analyze the evolutionary relationships of the Bellflower family (Campanulaceae), a group of small annual plants, many with attractive blooms (see Figure 3). We demonstrated a linear speedup with numbers of processors, essentially perfect parallel scaling [8], and a millionfold speedup compared to the original implementation of breakpoint analysis. The latest version of GRAPPA (version 1.6 released July 2002) includes significantly improved underlying algorithms for modeling the evolutionary process and reflects a billionfold speedup compared to the original 1998 breakpoint phylogeny algorithm.

GRAPPA is a prime example of the potential of high-performance algorithm development and HPC systems in computational biology. Such potential is likely to benefit researchers working on problems involving complex optimizations. Our reimplementation did not require new algorithms or entirely new techniques yet turned what had been an impractical approach into an effective one.

HPC and Next-Generation Biology

The HPC community has its roots in solving computational problems in physics (such as fluid flow, structural analyses, and molecular dynamics). However, traditional approaches to these problems, and to ranking HPC systems based on the Linpack

A HPC APPROACH IS NO SUBSTITUTE FOR INNOVATIVE ALGORITHM DESIGN *but is rather a natural complement.*

benchmark, may not be the optimal approach to HPC architectures in computational biology. Many researchers are carefully considering the architectural needs of HPC systems to enable next-generation biology. New HPC algorithms for biomedical research will require tight integration of computation with database operations and queries, along with the ability to handle new types of queries that are highly dependent on irregular spatial or temporal locality.

Many of the tools currently used in computational biology were created by biologists dealing with data sets that were miniscule in comparison to those available today. As a result, software that was once perfectly adequate now performs slowly or is incapable of successful analysis. As life scientists and biomedical researchers learn more about the complexities of protein structure, computational scientists find that the accurate simulation of a protein folding and changing its conformation due to biomolecular interactions may be intractable without PetaFLOPS-class computers. When algorithm engineering tools and practices are complemented by high-performance software implementations designed for parallel platforms, enormous gains will be realized in the size of data sets that may be analyzed and in the speed with which that analysis is carried out. GRAPPA is but one example of the benefits of this approach, which is likely to extend to a large variety of computationally intensive tasks.

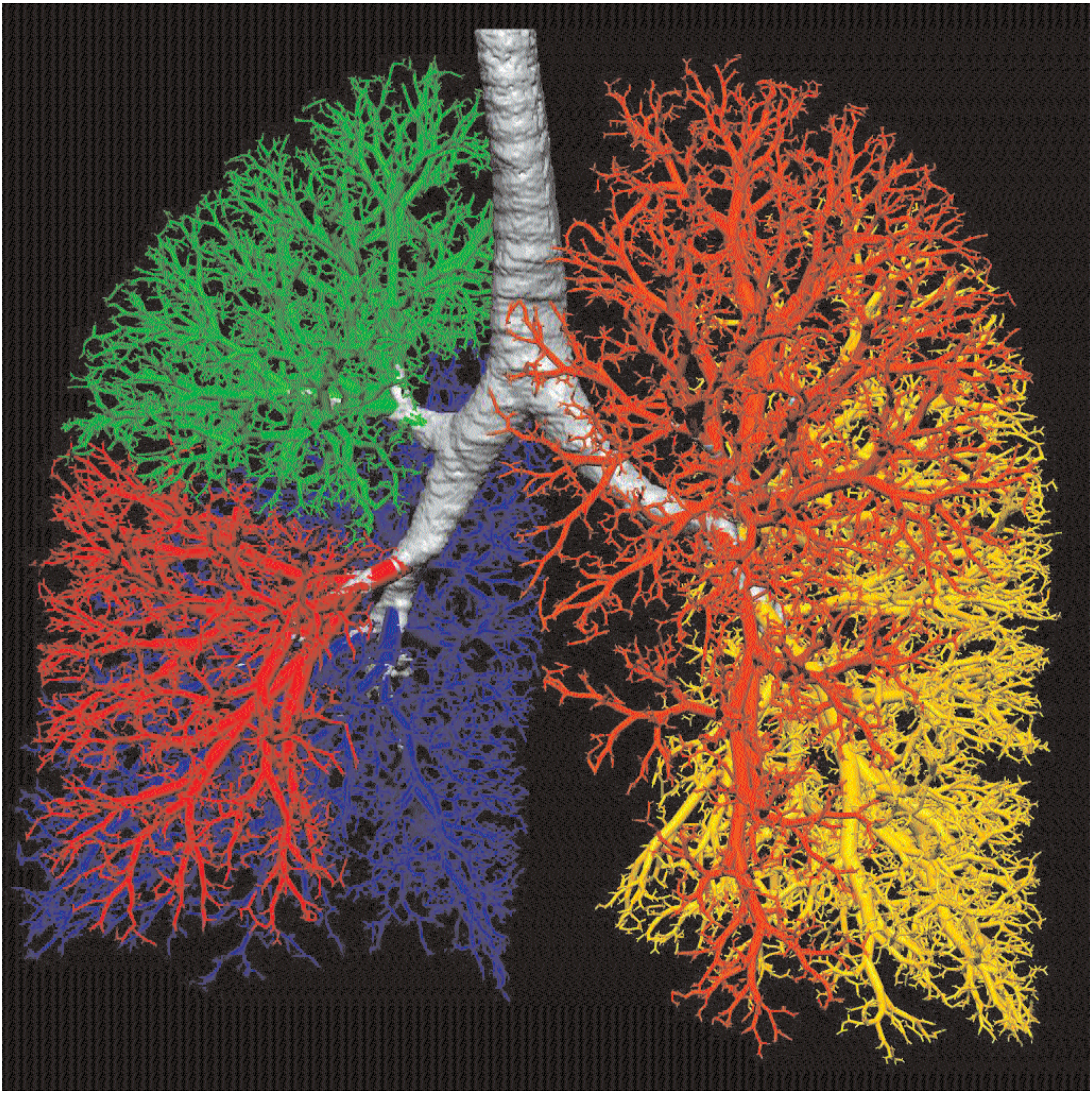
However, even large speedups have only limited benefits in theoretical terms when applied to NP-hard optimization problems. The billionfold speedup with GRAPPA allowed expansion of data sets from 10 taxa to 18 taxa. Thus, a HPC approach is no substitute for innovative algorithm design but is rather a natural complement. Much faster implementations, when sufficiently mature, might alter the practice of biomedical research. Research activities considered impossible due to computational challenges become feasible in theoretical biological research and applied biomedical research. Thus, approaches to scale and algorithmic design will enable HPC and biomedical researchers to solve today's grand challenge problems in both computing and biology. C

References

1. Bader, D., Moret, B., and Yan, M. A linear-time algorithm for computing inversion distance between signed permutations with an experimental study. *J. Comput. Biol. 8,* 5 (Oct. 2001), 483–491.
2. Dunker, A., Ensign, L., Arnold, G., and Roberts, L. Proposed molten globule intermediates in fd phage penetration and assembly. *FEBS Lett. 292,* 1–2 (Nov. 1991), 275–278.
3. Farris, J. The logical basis of phylogenetic analysis. In *Advances in Cladistics,* N. Platnick and V. Funk, Eds. Columbia University Press, New York, 1983, 1–36.
4. Felsenstein, J. Evolutionary trees from DNA sequences: A maximum likelihood approach. *J. Mol. Evol. 17,* 6 (Sept. 1981), 368–376.
5. Hannenhalli, S. and Pevzner, P. Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals). In *Proceedings of the 27th Annual Symposium on Theory of Computing* (Las Vegas, NV, May 29–June 1). ACM Press, New York, 1995, 178–189.
6. IBM Blue Gene Team. Blue Gene: A vision for protein science using a petaflop supercomputer. *IBM Syst. J. 40,* 2 (2001), 310–327.
7. Moret, B., Bader, D., and Warnow, T. High-performance algorithm engineering for computational phylogeny. In *Proceedings of the 2001 International Conference on Computational Science*, V. Alexandrov, J. Dongarra, and C. Tan, Eds. (San Francisco, May 28–30). *Springer-Verlag Lecture Notes in Computer Science 2073–2074* (2001), 1012–1021. Also in *J. Supercomputing 22,* 1 (May 2002), 99–111.
8. Moret, B., Wyman, S., Bader, D., Warnow, T., and Yan, M. A new implementation and detailed study of breakpoint analysis. In *Proceedings of the Sixth Pacific Symposium on Biocomputing 2001* (Big Island, HI, Jan. 3–7, 2001). World Scientific Publishing Co., Inc., Hackensack, NJ, 2001, 583–594.
9. Saitou, N. and Nei, M. The neighbor-joining method: A new method for reconstruction of phylogenetic trees. *Molec. Biol. Evol. 4,* 4 (July 1987), 406–425.
10. Sankoff, D. and Blanchette, M. Multiple genome rearrangement and breakpoint phylogeny. *J. Comput. Biol. 5,* 3 (Fall 1998), 555–570.
11. Schena, M., Shalon, D., Davis, R., and Borown, P. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science 270,* 5235 (Oct. 20, 1995), 456–460.
12. Venter, J. et al. The sequence of the human genome. *Science 291,* 5507 (Feb. 16, 2001), 1304–1351.

David A. Bader (dbader@ece.unm.edu) is an associate professor and Regents' Lecturer in the Departments of Electrical and Computer Engineering and Computer Science at The University of New Mexico in Albuquerque.

This work is supported in part by National Science Foundation grants CAREER ACI-00-93039, ITR ACI-00-81404, DEB-99-10123, ITR EIA-01-21377, Biocomplexity DEB-01-20709, and ITR EF/BIO 03-31654; and DARPA Contract NBCH30390004.



Finite element model for studying coupled air flow, blood flow, and soft tissue mechanics in the human lung.
Merryn Tawhai, The Bioengineering Institute at the University of Auckland, New Zealand.