Pacific Symposium on Biocomputing 6:238-239 (2001)

High-Performance Computing for Computational Biology

Thomas Ferrin
Departments of Pharmaceutical Chemistry and Biopharmaceutical Sciences
University of California
San Francisco, CA 94143-0446

Bruce A. Foster
Compaq Computer Corporation
High Performance Technical Computing
BioSciences and Chemistry
200 Forest Street MRO1-3/K14
Marlboro, MA 01752

Richard Hughey
Department of Computer Engineering
Jack Baskin School of Engineering
University of California
Santa Cruz, CA 95064

Computational biology has become a fully multidisciplinary field, including components of mathematics, biology, chemistry, and computer science. Computational biology is essentially the computer-aided analysis of the biology of organisms. Since even a single genome or proteome contains an immense quantity of data, performing even a simple analysis on genome-scale data quickly turns into a computationally difficult problem. Thus, computational biology now requires high-performance computing and its related components in database systems, visualization, and computer engineering.

The focus of this session is computational biology's growing need for high-performance computing. The first purely computational steps of the human genome project have required vast amounts of computing equipment with, for example, large processor farms being used in both the private and public assemblies of the human genome.

To the over-optimistic researcher, the simplest way to gain a factor of 10 performance increase is to use 10 processors. In practice, many critical issues degrade performance. Communication between the processing elements takes additional time, as does partitioning the problem and recombining the answers. Some algorithms cannot even be effectively partitioned. As seen in some of the papers in this session, often the most significant overhead is the additional time required to adapt an algorithm to a high-performance system.

The session received 26 papers, 11 of which the conference chairs selected for inclusion in this proceedings after evaluation by the session chairs and other reviewers. We were fortunate to receive a collection of papers that spanned most areas of practical high-performance computing. At the fine-grained level, we have discussions of dynamic programming sequence analysis on both a multithreaded processor (Martins *et al.*) and a specialized parallel coprocessor (Grate *et al.*) that is also used for computational chemistry (Rice and Hughey). Moving to clusters of computers, whole-genome alignment is performed using 65 microprocessors and a tree algorithm that includes both gaps and sequence inversions (Hsu and Cull). At the coarsest level, arbitrarily large collections of computers connected to the Internet can be viewed as a cluster for "Grid" computation. In one example, 80 remote processors sped binding-site searches (Waugh *et al.*).

Interactive visualization is a special category of high-performance computing, and poses distinct challenges due to the large volume of data that must be processed and the finely tuned algorithms that must closely complement the hardware in order to get real-time performance (Banatao *et al.*).

With the availability of high-performance computing, many researchers are pushing the bounds of practical computability with ever more sensitive and computationally costly algorithms. Three-dimensional multiple protein structure alignment is more demanding than the sequence alignment methods discussed above, but can be approached with high-performance computing and Monte Carlo optimization (Guda *et al.*). Another emerging area of research, whose computational needs can only grow, is the simulation of neural function (Dimitrov and Miller).

One of the most rapidly expanding areas of both experimental and computational biology has been the study of DNA microarrays. Here, we find two approaches to analyzing gene expression data. The first is a rule-based approach that combines expert annotation with machine learning techniques (Hvidsten *et al.*). The second uses a fully-automated probabilistic clustering algorithm to analyze expression data from the amoeba Dictyostelium discoideum (Sasik et al). Finally, in an impressive combination of coding theory and DNA microarrays, a means is found to verify the quality of the experimental process (Sengupta and Tompa).

We hope that this collection of papers will be helpful in your own work, as you struggle to find a balance between the complexity of your methods and the power of your computers.