

<http://DOEGenomeToLife.org/compbio/>

Report on the Computational Infrastructure Workshop for the Genomes to Life Program

**U.S. Department of Energy
Gaithersburg, Maryland
January 22–23, 2002**

**Workshop Organizers
Grant Heffelfinger, Sandia National Laboratories
Al Geist, Oak Ridge National Laboratory**

**Prepared by the Office of Advanced Scientific Computing Research
and
Office of Biological and Environmental Research
of the
U.S. Department of Energy
Office of Science**

March 2002

Table of Contents

Introduction	1
Importance of Computational Infrastructure to GTL Program.....	2
Summary Findings and Recommendations	2
Bioinformatics	3
Molecular Biophysics and Biochemistry	5
Modeling Complex Biological Systems	7
Application-Focused Infrastructure	9
Summary	10
Appendices	13
A: Workshop Attendees, January 2002	14
B: Agenda	15
Program-Planning Workshops for Genomes to Life	Inside back cover



Report on the Computational Infrastructure Workshop for the Genomes to Life Program¹

**U.S. Department of Energy
Gaithersburg, Maryland
January 22–23, 2002**



Introduction

Genomes to Life (GTL), a new program within DOE, seeks to identify and characterize the molecular machines of life, the gene regulatory networks that control them, and complex microbial communities. Cutting across these goals is the need to develop high-performance computational analysis and modeling capabilities and an infrastructure to support them.

DOE has long played a leading role in exploiting high-performance computing to accelerate advances in many scientific and application areas, including computational biology. Computational biology, however, has an unprecedented range of computing needs that make a well-planned infrastructure essential to achieving GTL's ambitious goals. These needs include the ability to perform informatics analysis on a diverse collection of distributed data sets produced by a variety of experimental methods, simulations that consume months of supercomputer time, and biological phenomena that no one yet knows how to model. The infrastructure for biology applications thus must not only provide high-speed computation for large-scale calculations but also must be compatible with much smaller scale calculations carried out on individual investigators' desktops. Such an infrastructure should be flexible, adaptable, and responsive to biology's evolving needs.

A workshop was held January 22–23, 2002, in Gaithersburg, Maryland, to analyze and document computational needs for the successful execution of the GTL program. The 34 attendees addressed questions in several key areas to identify the resources required and formulate a plan. This report provides a vision and some specific actions recommended to reach these goals.

The principal finding of workshop attendees was that only through computational infrastructure tuned and dedicated to the needs of biologists, coupled with new enabling technologies and applications, will it be possible to “move up the biological complexity ladder” and tackle the next generation of challenges.

¹This report was produced from the best available notes and does not represent a verbatim or consensus document of the workshop.

Importance of Computational Infrastructure to GTL Program

High-performance computing is essential to the type of high-throughput experimental biology that has emerged in the last 10 years. This was demonstrated by the success of the most visible such application to date—genomic sequencing. Sequence assembly and annotation have greatly extended the scale of bioinformatics and provided the incentive to establish a huge investment in and significant role for high-performance computing. Large computer farms have been established to provide capability for bioinformatics applications at numerous research institutions, including private companies, government laboratories, and universities. Computational needs for the next generation of challenges, however, may require a tighter coupling among processors.

As evidenced by GTL, biology is undergoing a major transformation that will be enabled and ultimately driven by computation. This can occur, however, only if an appropriate computational infrastructure is established. The data analysis and models required to understand molecular machines and microbial communities will become more computationally complex and heterogeneous and will require coupling to enormous amounts of experimentally obtained data. Such unprecedented problems can easily exceed the capabilities of next-generation (petaflop) super-computers. The following table presented at the workshop illustrates this point.

Problem	Computing Speed	Storage	Network
Genome Assembly	>10 Tflops to keep up with sequence rates	300 TB per genome	100 Mb/s
Protein Structure Prediction	>100 Tflops per protein set in a single microbe	Petabytes	500 Mb/s
Classical Molecular Dynamics	100 Tflops for each DNA protein interaction	10s of petabytes	2.4 Gb/s
First Principles Molecular Dynamics	1 Pflops per reaction in enzyme active site	100s of petabytes	10 Gb/s
Simulation of Biological Networks	>1 Tflops for small biological network	1000s of petabytes	unknown

Summary Findings and Recommendations

Emerging trends in three important computational biology areas—bioinformatics, molecular biophysics and biochemistry, and modeling complex systems—call for fundamentally new approaches to building a GTL computational infrastructure.

Bioinformatics

Perhaps the most challenging trend is the explosion of biological data. Massive and very complex, the body of data comes in different types and formats determined by experiments or simulations. It spans many levels of scale and dimensionality, including genome sequences, protein structures, protein-protein interactions, metabolic and regulatory networks, and multimodal molecular and cellular imagery. Existing biological data repositories are extremely dispersed, heterogeneous, and disintegrated, with various levels of intellectual property constraints.

A paradigm shift is needed in the development of bioinformatics infrastructure away from dispersed “data-collecting” repositories toward conceptually integrated “knowledge-enabling” repositories. Bioinformatics infrastructure must be more than storage and retrieval. Rather, in the long term, it must support fundamentally new ways of doing science. This view of “distributed resource management” is widely held, yet beyond the reach of any single research institution and very unlikely to be systematically addressed by any federal funding source besides DOE’s GTL program. For this reason, a major GTL success in this area would have tremendous impact on the biology community as a whole.

The need for new types of data infrastructure, a topic that emerged repeatedly during the workshop, was identified as a key challenge in achieving GTL goals. With large, complex biological databases and a diversity of data types, tools will be critical for accessing, transforming, modeling, and evaluating these massive data sets. Research groups must interact with these data sources in many ways. To be successful, the GTL infrastructure must provide users with cutting-edge data-management and data-mining software tuned to biology’s needs.

Group discussion emphasized a strong need to expand existing database technologies for better support of life science. Technically, GTL will need a database framework that supports and allows operations on new core data types natural to life science, has much richer features than are available in current databases, and performs reasonably well on typical life-science data.

Multiterabyte biological data sets and multipetabyte data archives will be generated by high-throughput technologies and petascale computing systems. The group emphasized that data-management issues must be addressed with high priority from the start of GTL. Among the issues are types of GTL-generated data, mechanisms for filtering data that needs to be stored, ways of disseminating data (publicly accessible, central vs dispersed repositories, federations), and mechanisms for capturing the data.

Types of data supported by databases should go beyond sequences and strings to include trees and clusters, networks and pathways, time series and sets, 3D models of molecules or other objects, shapes-generator functions, and deep images. Research is needed to allow for storing, indexing, querying, retrieving, comparing, and transforming those new data types. For example, such database frameworks should be able to index metabolic pathways and apply a comparison operator to retrieve all that are similar. Also, current



bioinformatics databases have very limited or no support for descriptions of simulations and large complex hierarchical models analogous to mechanical CAD or electronic CAD databases. Given the hierarchical nature of biological data, GTL databases should be able to organize biological data in terms of their natural hierarchical representations.

The group emphasized that having data standards would be ideal, but it is unrealistic to expect them soon. The key technical challenge toward this goal is to develop standardized semantics. Due to the complexity of biological data, its rapidly evolving nature, and problems with synonymy (different names with the same meaning) and polysemy (the same name for different concepts), standards tend to be several steps behind. For this reason, the group concluded that using temporary standards and continuing efforts would be important in merging standards among multiple groups with such similar domains as metabolic pathways and networks.

GTL will need a flexible data-management framework because technology used in biology is changing at a fast pace. Data types will be determined by new experiments, analyses, and simulations. These in turn will impact infrastructure requirements. Data-storage strategies thus should be allowed to evolve over time in an organized, timely, and economical way. Much care should be given to evaluating tradeoffs between storing and regenerating modeling and analysis data.

Computational analysis of data is a key component of GTL (and systems biology in general), and there is a critical need for tools and tool frameworks that allow biologists to derive inferences from massive amounts of heterogeneous and distributed biological data. Data-analysis infrastructure should support an environment for creating and managing sophisticated, distributed data-mining processes. Using intuitive visual interfaces, developers and data analysts should be able to program new data-mining applications or open existing application templates that easily can be customized to a given problem's unique requirements. Such processes should have both application and Web-based streamlined interfaces. An infrastructure should encompass a large repository of analysis modules including analyses of sequences, gene expression, phylogenetic tree, and mass spectrometry.

Among other features, the discussion group listed the need to incorporate support for probabilities and confidence factors into databases, making visualization a part of database infrastructure, and providing "query-by-example" capabilities. Databases should drive other workflow aspects of problem-solving tasks. For example, they should be part of simulation environments in which all model parameters and elements are in the database and the user can construct a simulation on the fly. Databases should have features for deductions and knowledge structures that would allow for automatic inference in response to a query, given some axiomatic rule-based understanding of qualitative and quantitative laws in biology.

Databases need robust interfaces to such experimental systems as chips, detectors, microscopes, and mass spectrometers. They should be able to support workflow and experimental planning as part of a controlled workflow-management system. Finally, the performance of these large-scale databases must be robust, scalable, and efficient enough for use by the broad life-sciences community.



The group emphasized the need to develop an easy-to-use search infrastructure that addresses the scale, heterogeneity, and distributed nature of biological data. Such an infrastructure should enable search services to operate across domains by providing user-configurable tools for mapping between metadata schemas and performing search queries before and after processing and against multiple data sources.

In general the group felt that, in addition to satisfying immediate production requirements, a data infrastructure should have built-in flexibility to accommodate longer-term GTL needs.

Finally, the group concluded that the data-infrastructure effort is too large to be solved independently within any single program. Researchers supported by the DOE Office of Biological and Environmental Research and the Office of Advanced Scientific Computing Research should work closely on these data-related issues. In particular, GTL should leverage the tools and intellectual output of the Scientific Discovery Through Advanced Computing (SciDAC) program.

Bioinformatics applications often are trivially parallel. Thus, the hardware and operating-system requirements for bioinformatics are less about flop-rates and interprocessor communication speeds and more about parallel input and output between processors and memory. Successful bioinformatics tools should enable life-science researchers to seamlessly link data (often geographically distributed via the Internet) with modeling and simulation results.

Recommendations

Establish a data infrastructure to make the growing body of biological data available in a form suitable for study and use by developing and implementing:

- Methodologies necessary for seamless integration and interoperation of distributed computational and data resources, linking both experiment and simulation.
- Life-sciences-enabling database frameworks that provide complex and multidatabase queries, new data models natural to life science, enhanced operations on these data types, and optimized performance.
- Data-analysis and interpretation systems that will enable database transition from data-collecting repositories to information and knowledge bases. Such databases will provide inference capabilities for establishing relationships across multiple sources of information (genomic sequence, gene-protein expression, protein-protein interactions, protein structures and complex structures, and biological pathways) leading to new scientific discoveries.

Molecular Biophysics and Biochemistry

New and existing modeling and simulation methods are emerging rapidly as tools to explain and understand biological data and phenomena. This is especially true in molecular biophysics and molecular biochemistry. For some problems, existing simulation meth-



ods are up to the task. Rapid in silico screening of drug candidates with such methods as molecular modeling, structure-based design, and virtual screening is beginning to make significant contributions. In other cases, complex interactions and molecular phenomena with long time scales remain beyond the reach of existing molecular simulation.

Although they are important for relating protein structure to function for GTL-relevant subcellular processes, these methods presently are best applied in conjunction with high-throughput experiments followed by bioinformatics analysis and top-down modeling of complex cellular processes. There are several reasons. First, molecular simulation consumes an enormous amount of computational resources and therefore is best focused on specific molecular biology problems. Second, given the complement of hundreds of thousands of proteins in a single cell, a bottoms-up or reductionist approach is far beyond the computational capabilities currently available to life-science researchers. Thus, molecular simulation has two major roles in the GTL program: (1) link structure and function for focused molecular machines, and (2) provide physical and chemical understanding of specific molecular processes identified as crucial to the complex subcellular processes that regulate cellular response.

Molecular simulation requires massively parallel supercomputers with high-speed interconnects to solve a single problem. Furthermore, the task of achieving 1000-fold speed-up on 1000 processors for a single application code requires the development of efficient new parallel algorithms. DOE laboratories have been pioneers in this area since massively parallel computers were first conceived.

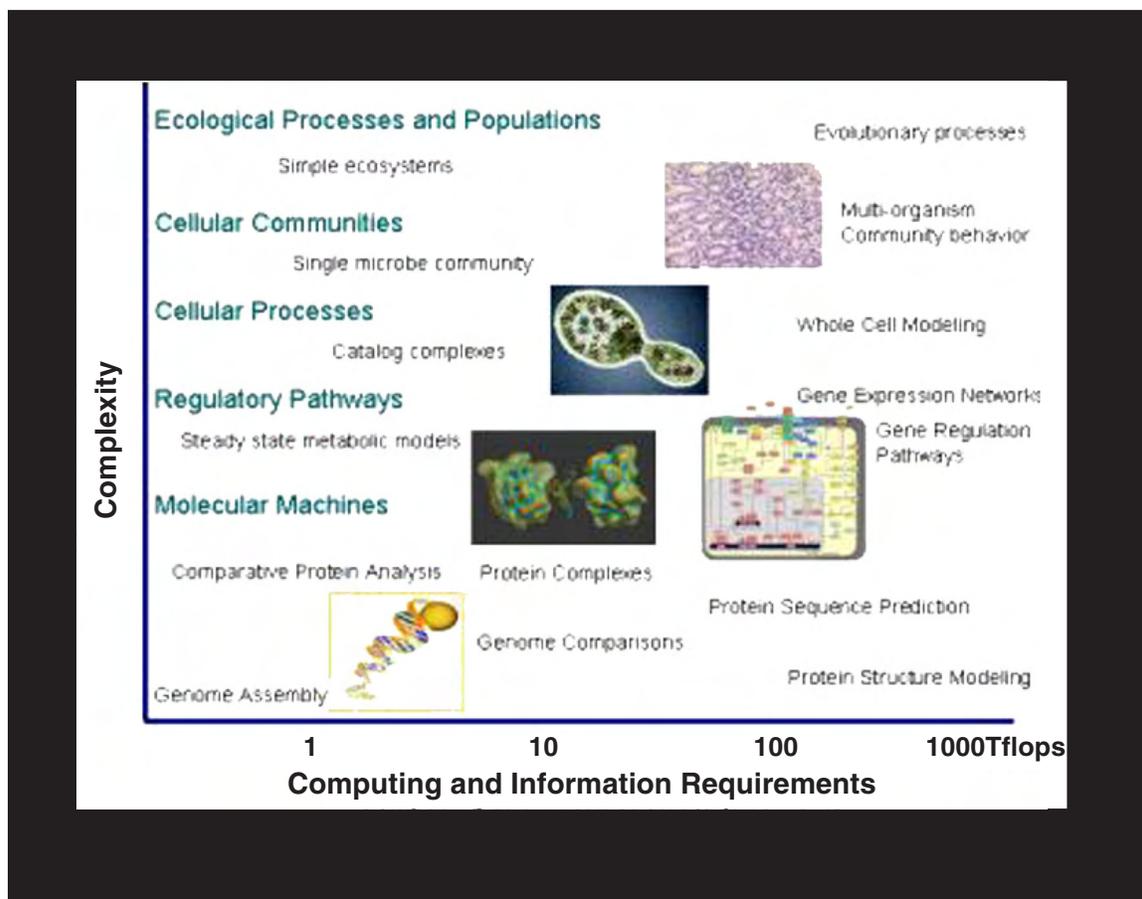
New molecular-simulation methods and companion algorithms for molecular biology questions specifically relevant to GTL goals are an important aspect in developing the needed GTL computational infrastructure. Furthermore, because these applications require careful attention to microprocessor flop rates as well as interprocessor communication speeds, more attention needs to be given to hardware and operating systems to produce a computational infrastructure that can efficiently model biological systems.

Recommendations

- Molecular physics and chemistry methods will be most useful to GTL researchers in understanding machines such as protein complexes. For this reason, the computational requirements of these methods, both existing and anticipated, should be considered in evaluating GTL program needs.
- Molecular simulation is best applied to specific problems where molecular understanding is desired, rather than as part of a large-scale effort to extend fundamental molecular insight to the cellular system level (“the reductionist approach”).
- Molecular-simulation tools should be considered useful components of a computational biology capabilities toolkit that connects high-throughput experiments to models of molecular structure and function.

Modeling Complex Biological Systems

Modeling complex biological systems will require new methods to treat the vastly disparate length and time scales of individual molecules, molecular complexes, metabolic and signaling pathways, functional subsystems, individual cells, and, ultimately, interacting organisms and ecosystems. Such systems act on time scales ranging from microseconds to millions of years. These enormously complex and heterogeneous full-scale simulations will require not only petaflop capabilities but also a computational infrastructure that allows for model integration while simultaneously coupling to huge databases created by an ever-increasing number of high-throughput experiments. Using high-end computing to address scientific questions is an unprecedented challenge for the computational science research community. See the figure below.



To create extremely heterogeneous cellular models, the modeling of biological systems needs to be supported at various levels of abstraction. Some possible layers may be modeling at the sequence level with different regulation schemes; at the levels of molecular machines, molecular complexes, networks and pathways (including metabolic, signaling, and regulation); at the structural components level that incorporates spatial organization of the cell, cell structure, and morphology; in modeling such extracellular



environments as biofilms; and ultimately in modeling populations and consortia. Research is needed to develop robust interfaces for coupling these modeling levels and abstracting from one layer to the next. Thus, this infrastructure is critical to understanding how to build a software environment that would allow construction of user-driven simulations rather than focusing on a specific simulation project.

The group emphasized the need to develop an infrastructure that would facilitate the fast transition of algorithms from papers into tools available to an average person sitting in the laboratory. Toolkits resembling “Mathematica” or “Matlab” for molecular, cellular, and systems biology might be one of the components in this infrastructure.

Such an infrastructure will require building core models and underlying structures with very high performance implementations of fundamental data objects, including general-purpose integers and arbitrary precision floating points as well as objects specific to molecular systems biology such as trees, clusters, and networks. The infrastructure should have a general set of optimized core library functions, including algorithms for restriction maps and map assembly (planning cloning and clone libraries, building physical genome maps); modules for sequence assembly and multiple sequence assembly (data models and sequence-analysis algorithms, multiple sequence alignment, probability and statistics for sequence alignment and patterns, gene prediction, mutation analysis); modules for trees and sequence comparisons and construction (phylogenetic tree construction and analysis, comparative genomics); and modules for proteomics analysis (protein structure prediction and kinetics prediction, array analysis).

These modules should be embedded in a script-driven environment to permit rapid prototyping and interfaced with database systems that have built-in schemas for representing common tasks. They should be pluggable into high-end simulations, have parallel and accelerated kernels to exploit the massively parallel computers that will be part of the GTL infrastructure, and have ties into collaborative workflow and group interfaces for teaching and collaboration.

Because their computational requirements are so diverse, coupling informatics with modeling and simulation establishes the need for a fully general-purpose computing infrastructure. Hardware needs for such a challenge range from commodity clusters to tightly coupled, massively parallel architectures with greater investment in interprocessor communication. Implications for operating systems are equally disparate, requiring in some cases extremely high rates of parallel input-output to move data among processors and memory as well as efficient management of single-application codes distributed over hundreds or thousands of processors.

Recommendations

- Develop and implement a hierarchical modeling environment that provides robust interfaces for multiple levels ranging from sequence through structure to extracellular environments and ultimately populations and consortia.

- Develop and implement efficient and interoperable computational tools via new software technologies including “Mathematica”-type toolkits for molecular, cellular, and systems biology. These tools should have highly optimized core life-science library modules embedded into script-driven environments for rapid prototyping and be interfaced with database systems, pluggable into high-end simulations, and tied into collaborative workflow for teaching and collaboration.

Application-Focused Infrastructure

The combined implications of these disparate computing approaches and complex system modeling are unique to biology. Although some individual pieces of the integrated vision discussed above have been implemented on teraflop-scale computers and in some cases optimized for different platforms, the next generation (petascale) of life-science codes will be running in computing environments far more complex than those commonly used by biological researchers today. Furthermore, computational infrastructures will not appear without advance planning to make these systems easy to use and optimized for delivering a sustained hardware peak performance on biology applications with widely disparate computational requirements.

Biologists should embrace high-performance computing as a tool, and the computational infrastructure needs to occur at both the software and personal levels. This can be facilitated by building a biological science network that connects computing and human resources for experiment, discovery, education, and teaching and ensures timely access and interactive teamwork-driven problem-solving. There have been only limited amounts of such integration in the past, but GTL will be successful only if much more attention is paid to considerations including the following:

- Integration of modern enabling technologies with legacy and developing biological applications.
- Collaboration between computational scientists and biologists on how best to exploit and utilize high-performance computing resources.

The group discussed the creation of a biological science network connecting various resources, including people (biologists, computer scientists, mathematicians); experimental systems (arrays, detectors, MS, MRI, EM); databases (data centers, curators, analysis servers); simulation resources (supercomputers, visualization, desktops); discovery resources (e.g., search servers for optimized hardware); and education and teaching resources (classrooms, laboratories). They also discussed how this network would be different from current computer science “grid” projects because there is much more integration on the lab-level scale and most participants will be experimentalists. This network would need to be more diverse in data sources and databases (e.g., integration, federation) than any current data grids. The conclusion was that such a biological science network would be developed in directions distinct from and complementary to the “grid” currently pursued by the computer science community.

Workshop attendees raised points that cut across the entire GTL program. A serious issue is the movement of researchers to industry. To sustain such a large program, steps should be taken to retain and increase the number of researchers with the required expertise. Another general concern was that software developed as part of GTL would be unavailable to other researchers. Software and data often are held by researchers today until they publish papers. Supporting the release and maintenance of software is important for GTL to consider if the larger computational biology community is to benefit from the program's output.

Recommendations

A biological science network approach, with sponsor support, could

- Encourage the same model of open source software distribution being used in the DOE Scientific Discovery Through Advanced Computing initiative, thus leveraging the discussions and solutions of SciDAC.
- Support the conversion of prototype software systems to production-grade versions when their user base becomes large. Since many users may be experimentalists, DOE also should consider creation of user-support services for GTL software.
- Increase the number of scientists with biocomputing and bioinformatics expertise. DOE should support sufficient scope of multidisciplinary research, training, and outreach that will be necessary for GTL's success. DOE should periodically sponsor workshops, symposia, and tutorials so that a broad community will be trained to use existing biocomputing tools and stimulated to contribute to the development of new tools.
- Work in partnership with other government agencies, to establish mechanisms for adequate funding of purchasing, maintenance, and upgrades of the GTL computational infrastructure.
- Broadly disseminate GTL achievements, including research results, data, and software tools.

Summary

GTL needs to be more than the sum of independent projects bolted together. It must have an infrastructure in which collaborators at multiple sites can interact and have access to data, high-performance computation, and storage resources. Establishing this infrastructure will involve many technical challenges, including the following:

- Creation of user-friendly tools with transparent utilization of high-performance computers and distributed databases.
- Distributed analysis of ever-increasing databases of diverse biological data for inclusion into models.
- Effective database design and database query in support of modeling.

- 
- Integration of models into problem-solving environments and incorporation of data to determine simulation parameters and validate results.
 - Streamlined data from such experimental devices as mass spectrometers, NMR systems, and light and neutron sources into both databases and models.
 - Creation of network and storage infrastructure to handle next-generation bioinformatics needs.
 - Bringing both experimental and modeling groups together in collaborative environments.
 - Coupling of whole-cell models on petaflop-scale systems with smaller component models on workstations.
 - Seamless end-user access to applications, data storage, and computer resources to support high-end modeling.

Appendices

Appendix A: Workshop Attendees, January 2002

Appendix B: Agenda

Appendix A: Workshop Attendees, January 2002

Speakers:

Gary Johnson U.S. Department of Energy
John Wooley University of California, San Diego
Marshall Peterson Celera Genomics
Bill Camp Sandia National Laboratories
Bill Beavis National Center for Genome Research
Steve Wiley Pacific Northwest National Laboratory
Rick Stevens Argonne National Laboratory

Data Panel

Ying Xu Oak Ridge National Laboratory
Terence Critchlow Lawrence Livermore National Laboratory
Carl Anderson Brookhaven National Laboratory

Organizers

Al Geist Oak Ridge National Laboratory
Grant Heffelfinger Sandia National Laboratories
Nagiza Samatova Oak Ridge National Laboratory
Mike Colvin Lawrence Livermore National Laboratory
Ray Bair Pacific Northwest National Laboratory
Esmond Ng Lawrence Berkeley National Laboratory

Other attendees

Steve Plimpton Sandia National Laboratories
Mark Sears Sandia National Laboratories
Natalia Maltsev Argonne National Laboratory
Ed Uberbacher Oak Ridge National Laboratory
Rodger Brent Molecular Sciences Institute
Eugene Kolker Institute for Systems Biology
Andrew Fant Vertex Pharmaceuticals
David Deerfield Pittsburgh Supercomputing Center
Horst Simon National Energy Research Scientific Computing Center
Bill Kramer National Energy Research Scientific Computing Center
Thomas Zacharia Oak Ridge National Laboratory
Gyan Bhanot IBM
Jim Leightner Energy Sciences Network
George Seweryniak U.S. Department of Energy
Walt Polansky U.S. Department of Energy
Fred Johnson U.S. Department of Energy
John Houghton U.S. Department of Energy
Sylvia Spengler National Science Foundation

Appendix B

Final Agenda Computational Infrastructure for the Genomes to Life Program

January 22-23, 2002

Gaithersburg Hilton, Gaithersburg, Maryland

Organizers: Grant Heffelfinger, Sandia National Laboratories
Al Geist, Oak Ridge National Laboratory

There are 15 minutes between each speaker for questions and discussion plus a group discussion period at the end of each session.

January 22, 2002

8:30–12:00 Hardware Infrastructure

Overall vision: John Wooley, University of California, San Diego

Hardware from Bio perspective: Marshall Peterson, Celera Genomics

Bio from CS perspective: Bill Camp, Sandia National Laboratories

Group discussion

12:00–1:00 Working Lunch

1:00–4:30 Enabling Technologies

Overall Vision: Bill Beavis, National Center for Genome Research

Enabling Technologies
from Bio Perspective: Steve Wiley, Pacific Northwest National Laboratory

CS perspective: Rick Stevens, Argonne National Laboratory

Group discussion

January 23, 2002

8:30–12:00 Data and Networking Infrastructure

Panel: Ying Xu, Oak Ridge National Laboratory
Terence Critchlow, Lawrence Livermore National Laboratory
Carl Anderson, Brookhaven National Laboratory

Group discussion: Bio needs in data and networking and initial discussions of a “Data Standard”

12:00 Meeting ends

