

**AFRL-IF-RS-TR-2005-160**  
**Final Technical Report**  
**April 2005**



# **BIOMEDICAL REQUIREMENTS FOR HIGH PRODUCTIVITY COMPUTING SYSTEMS**

**Federation of American Scientists**

**Sponsored by**  
**Defense Advanced Research Projects Agency**  
**DARPA Order No. N353**

*APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.*

**The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.**

**AIR FORCE RESEARCH LABORATORY**  
**INFORMATION DIRECTORATE**  
**ROME RESEARCH SITE**  
**ROME, NEW YORK**

## **STINFO FINAL REPORT**

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2005-160 has been reviewed and is approved for publication

APPROVED:                   /s/  
CHRISTOPHER J. FLYNN  
Project Engineer

FOR THE DIRECTOR:                   /s/  
JAMES A. COLLINS, Acting Chief  
Advanced Computing Division  
Information Directorate

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                     |                                                                    |                                                                                                     |                                   |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|--------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|-----------------------------------|
| <b>REPORT DOCUMENTATION PAGE</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                                     |                                                                    | Form Approved<br>OMB No. 074-0188                                                                   |                                   |
| Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503                                                                                                                        |                                                                     |                                                                    |                                                                                                     |                                   |
| <b>1. AGENCY USE ONLY (Leave blank)</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                     | <b>2. REPORT DATE</b><br>April 2005                                | <b>3. REPORT TYPE AND DATES COVERED</b><br>Final<br>May 02 – Apr 04                                 |                                   |
| <b>4. TITLE AND SUBTITLE</b><br><br>BIOMEDICAL REQUIREMENTS FOR HIGH PRODUCTIVITY COMPUTING SYSTEMS                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                     |                                                                    | <b>5. FUNDING NUMBERS</b><br>G - F30602-02-1-0117<br>PE - 62712E<br>PR - BR4H<br>TA - PC<br>WU - 04 |                                   |
| <b>6. AUTHOR(S)</b><br>Kay Howell<br>Gerry Higgins                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                     |                                                                    |                                                                                                     |                                   |
| <b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b><br><br>Federation of American Scientists<br>1717 K St., NW<br>Suite 209<br>Washington DC 20036                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                     |                                                                    | <b>8. PERFORMING ORGANIZATION REPORT NUMBER</b><br><br>N/A                                          |                                   |
| <b>9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b><br><br>Defense Advanced Research Projects Agency<br>3701 North Fairfax Drive<br>Arlington VA 22203-1714                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                     |                                                                    | <b>10. SPONSORING / MONITORING AGENCY REPORT NUMBER</b><br><br>AFRL-IF-RS-TR-2005-160               |                                   |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                     |                                                                    | AFRL/IFTC<br>525 Brooks Road<br>Rome NY 13441-4505                                                  |                                   |
| <b>11. SUPPLEMENTARY NOTES</b><br><br>AFRL Project Engineer: Christopher J Flynn/IFTC/(315) 330-3249 Christopher.Flynn@rl.af.mil                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                                     |                                                                    |                                                                                                     |                                   |
| <b>12a. DISTRIBUTION / AVAILABILITY STATEMENT</b><br><br>APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                     |                                                                    |                                                                                                     | <b>12b. DISTRIBUTION CODE</b>     |
| <b>13. ABSTRACT (Maximum 200 Words)</b><br>This report details biomedical computing requirements for high productivity computing systems for DARPA's High Productivity Computing Systems (HPCS) program. The goal of the project was to determine biomedical computing requirements in order to define the size and nature of the demand in this research field; provide an assessment of the impact HPCS technologies can have on important biomedical problems and highlight HPCS R&D areas critical to advances in biomedical computing. The study team used multiple techniques to gather, assimilate and validate biomedical computing requirements including: a review of biomedical computing needs in the literature, interviews with government and industry researchers, and a workshop to identify software environment requirements. |                                                                     |                                                                    |                                                                                                     |                                   |
| <b>14. SUBJECT TERMS</b><br>High productivity computing, Biomedical computing requirements                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                     |                                                                    |                                                                                                     | <b>15. NUMBER OF PAGES</b><br>122 |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                     |                                                                    |                                                                                                     | <b>16. PRICE CODE</b>             |
| <b>17. SECURITY CLASSIFICATION OF REPORT</b><br><br>UNCLASSIFIED                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | <b>18. SECURITY CLASSIFICATION OF THIS PAGE</b><br><br>UNCLASSIFIED | <b>19. SECURITY CLASSIFICATION OF ABSTRACT</b><br><br>UNCLASSIFIED | <b>20. LIMITATION OF ABSTRACT</b><br><br>UL                                                         |                                   |

## **Abstract**

This report details the results of a study to determine biomedical computing requirements for high productivity computing systems. The report examined needs with regards to: 1) the size and nature of demand; 2) the potential impact of high productivity computing technologies on important biomedical applications; and 3) R&D areas critical to advances in biomedical computing. The report was compiled using multiple techniques, including: a review of biomedical computing needs as reported in literature; telephone and personal interviews with researchers and program managers; and a workshop to identify software environment requirements. The study focused on five research areas: bioinformatics; computational protein biochemistry; computational biology; drug discovery; and computer-aided diagnostic imaging and image-guided interventions, and included case studies of researchers.

## Table of Contents

|                                                                                   |     |
|-----------------------------------------------------------------------------------|-----|
| Abstract .....                                                                    | i   |
| Introduction and Summary: .....                                                   | 1   |
| Chapter 1: Biomedical Computing Overview .....                                    | 2   |
| Chapter 2: Bioinformatics .....                                                   | 8   |
| Chapter 3: Computational Protein Biochemistry .....                               | 23  |
| Chapter 4: Computational Biology .....                                            | 33  |
| Chapter 5: Drug Discovery .....                                                   | 56  |
| Chapter 6: Computer-Aided Diagnostic Imaging and Image-Guided Interventions ..... | 68  |
| Chapter 7: Virtual Soldier Project.....                                           | 80  |
| Chapter 8: Interviews.....                                                        | 86  |
| Conclusion .....                                                                  | 112 |
| References .....                                                                  | 113 |

## List of Figures

|                                                                                                              |    |
|--------------------------------------------------------------------------------------------------------------|----|
| Figure 1.1: Biological Complexity.....                                                                       | 4  |
| Figure 1.2: Complexity and Timescale.....                                                                    | 5  |
| Figure 2.1: Growth of GenBank .....                                                                          | 10 |
| Figure 7.1: Parallelization of Material Point Method (MPM) Codes for Torso and<br>Cardiovascular Wounds..... | 83 |
| Figure 7.2: Schematic of a Single Computational Step in MPM Algorithm .....                                  | 84 |

## **Introduction and Summary:**

This report details the results of the Biomedical Computing Requirements for High Productivity Computing Systems project for DARPA's High Productivity Computing Systems (HPCS) R&D program. The biomedical HPCS community is a key segment of the industrial user community, important because of its potential market size and because of the large public-health benefits that can result from advances in medical research enabled by high-end computing. The research goal of this project is to determine biomedical computing requirements for high productivity computing systems in order to (1) define the size and nature of demand, (2) provide an assessment of the impact high productivity computing systems (HPCS) technologies can have on important biomedical problems and (3) highlight HPCS R&D areas critical to advances in biomedical computing.

The study team used multiple techniques to gather, assimilate, and validate the biomedical HPCS requirements, including: (1) a comprehensive review of biomedical computing needs as reported in current literature; (2) telephone interviews with researchers and program managers; (3) a workshop to identify software environment requirements; (4) assimilation of the findings from tasks (1) - (3) into a preliminary report; (5) a workshop at which the preliminary report was discussed and revised as needed; and (6) a final report that synthesizes the discussions from the workshop and other community input.

## Chapter 1: Biomedical Computing Overview

Biomedical computing is the application and development of computer methods for biomedical research<sup>1</sup>. It spans many disciplines including bioinformatics, molecular modeling, systems biology, medical imaging, and others. The ultimate goal of biomedical computing is to advance the biomedical sciences by simulating life at all applicable levels of detail—the biochemical, physiological, cell, organ, organism, and population levels. The results promise to include better diagnoses, better drugs and other therapies that are developed faster, perhaps through mass customization, better surgical procedures, better prostheses, better recognition and repair of public health problems, and, thus, a healthier population, perhaps with lower medical costs.

**Example:** In collaboration with the Cleveland Clinic Foundation and the University of Auckland in New Zealand, the researchers in the Cardiac Mechanics Research Group explore the potential of a revolutionary surgical method for patients with severe heart failure. Through the combination of computational modeling with magnetic resonance imaging, the research is to predict which patients effectively can be rescued, using surgical ventricular reduction.

Computers were originally invented to address problems in physics and cryptanalysis. Later various communities realized the applicability of generalized computing to their applications. But the alignment of applications with computing capability has often lagged the available capability. For example, widespread use of computers in the first business applications occurred in the late 1950's through the 1960's, perhaps 12-15 years after the availability of adequate hardware. This is because application practitioners are typically focused more on the nature and progress of the applications *per se* than on the technology available to make processes more efficient. Typically computer systems have been adopted to speed up existing processes; eventually, enough practitioners arise who fully understand the nature of both the application area and the computing environment to enable the leap to applications that never would have been possible without the computer. This has occurred in the physical sciences and many communications, business and entertainment application areas, among others. The life sciences lag the physical sciences somewhat in this evolution, but such applications as 3-D computed

tomography and other imaging, genetic analysis, and advanced simulations hint at the dramatic possibilities for accelerating scientific advances and at a hugely expanding market for biomedical computing. Some of these applications require the most advanced computing capabilities available, and, indeed, some demand advances in the state of the computing technology at all levels.

As has happened previously in the physical sciences, the role of computing is dramatically increasing in all areas of biological research. The initial wave of computational use focused on sequence analysis. While many unsolved problems remain in sequence analysis, current and future needs will focus on integration of diverse sets of data, originating from a variety of experimental techniques which are capable of producing data at the levels of entire cells, organs, organisms and populations<sup>2</sup>.

**Example:** A computational model of the cardiovascular system is aiding researchers in understanding the fundamental biochemical, biophysical, electrical and mechanical functions of the normal heart. The model is also advancing understanding of the molecular and genetic origins of heart disease, the electrical and mechanical properties of blood flow in large and small blood vessels; and the development of potential approaches for new cardiovascular drugs. A virtual lung model, developed at the Department of Energy's Pacific Northwest National Laboratory, may help predict the impact of pollutants on respiratory systems and provide new insights into asthma, as well as other pulmonary diseases<sup>3</sup>. Using the virtual respiratory tract, PNNL scientists can analyze the influence of various factors, such as the amount of pollutants or length of exposure, on healthy versus diseased lungs by manipulating the computer model. With the model they can begin to simulate how gases, vapors and particulates may act differently within lungs of people suffering from cystic fibrosis, emphysema and asthma.

Biomedical computing presents many challenges. First, biology is inherently non-linear and complex – current models are simplified linear approximations, often developed to fit into available computing resources. Inaccuracies entailed by this linearity severely limit the models' applicability.

Another challenge is the sheer size of the solution spaces for some problems that must be solved by searching. For example, the alignment of two sequences of length 100 has on the order of  $10^{30}$  possible solutions. Various search strategies are employed to narrow or jump around the space, but the problem is still very hard to compute. The following chart demonstrates the relationship between complexity and timescales.

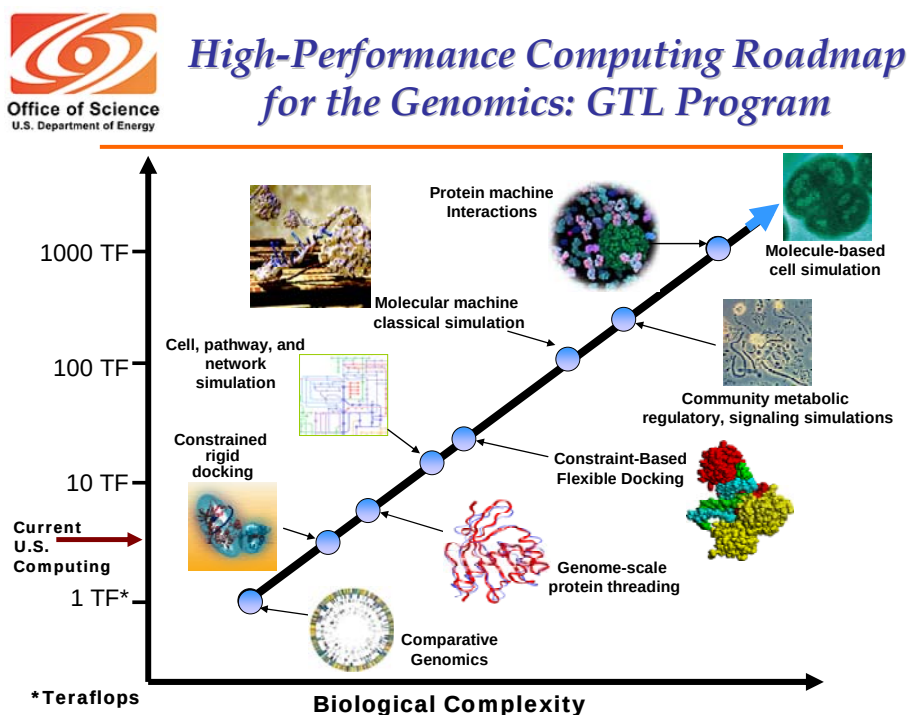


**Figure 1.1 Biological Complexity.**

Source: David Thomassen, Office of Biological and Environmental Research, DoE Office of Science

A third major challenge is system complexity and the need to span multiple scales of biological organization. The dimensions of biological interest range from small organic molecules to multi-protein complexes to cellular processes to tissues to the interaction of human populations with the environment. Timescales present yet another challenge – times can vary from microseconds to generations of populations. The time scales of biological function range from very fast femtosecond molecular motions, to multi second protein folding pathways, to cell cycle and development processes that take place over the order of minutes, hours and days. The linking of biological phenomena at all levels of temporal and spatial scale is driving the transformation from separate, anatomically-based domains of biological research to systems level research.

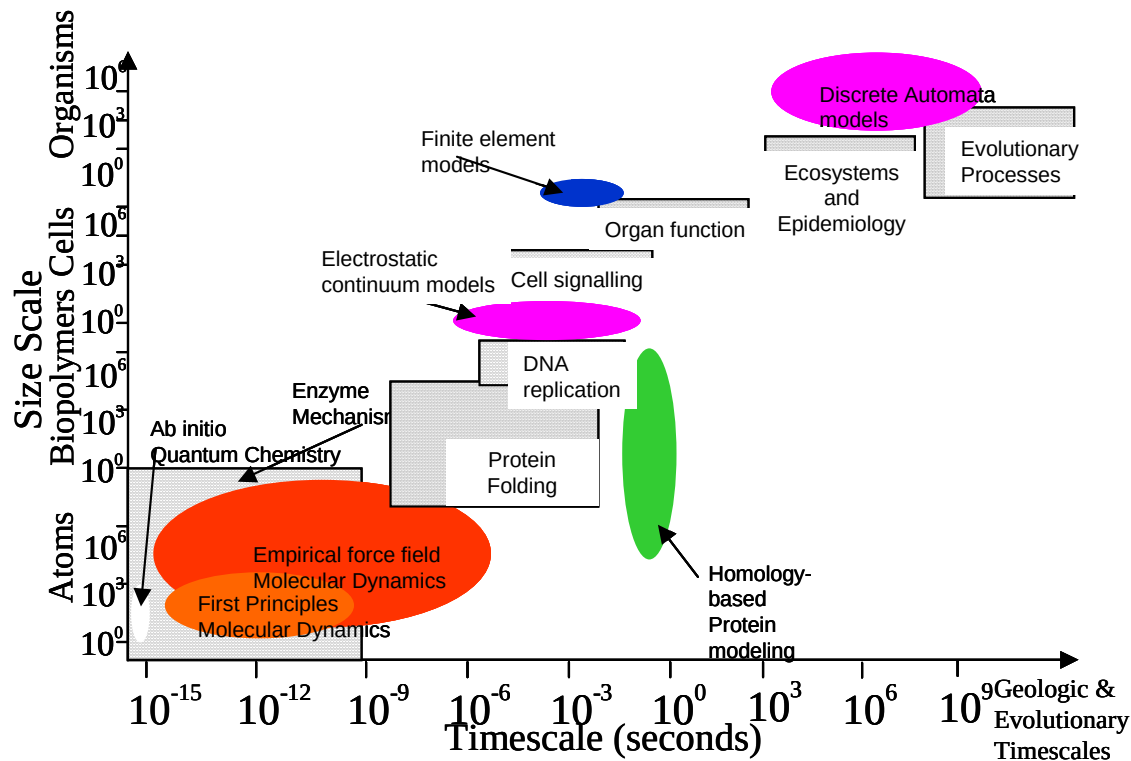


**Figure 1.2 Complexity and Timescale**

This transformation is, in turn, creating a critical need for theoretical, algorithmic and software advances in storing, retrieving, networking, processing, analyzing, navigating and visualizing biological information. Indeed, biomedical computing is in many ways in its infancy in regards to use of computing; this presents its own set of challenges in that the field lacks well established algorithms and computational methods. In addition, while computing capabilities increase continuously, biomedical computing models generally are not re-designed to take advantage of either new hardware or of the most advanced high-end systems.

The inherent complexity of biological systems, resulting from biological evolution and our lack of a comprehensive theory of biological organization at the molecular level requires sophisticated machine learning approaches in order to deal with huge amounts of data. Machine learning methods (neural networks, hidden Markov models, etc.) are well suited for domains characterized by large quantities of data, noisy patterns and the absence of general theories.

These methods are computationally intensive, clearly require high-end computing capabilities, and would benefit from further improvements in computational performance.

Through interviews to date, the study team has identified the following biological research where the requirement for HPCS is already widely recognized:

- Structure of proteosome, ribozyme, ribosome, Adenosine Triphosphatase (ATPases), Virus, membrane protein complexes
- Whole genome comparison
- Combined quantum/classical simulations
- Protein folding/threading
- Microsecond time-scale simulations
- Protein-protein and protein-DNA recognition and assembly

This report details the results of the Federation of American Scientists' review of Biomedical Computing Requirements for High Productivity Computing Systems under the project for DARPA's High Productivity Computing Systems (HPCS) R&D program. We focus on five research directions and, where available, present case studies of a researcher within that thrust area.

## 1.1 Bio-Computing Research Directions

This report organizes biomedical computing in five major categories, each with a review of its algorithms and computational methods. The categories are as follows:

- **Bioinformatics** (includes genomics, DNA sequencing, microarray technologies and bioinformatics): Research, development, or application of computational tools and approaches for expanding the use of biological, medical, behavioral or health data, including those to acquire, store, organize, archive, analyze, or visualize such data.
- **Computational Protein Biochemistry** (includes protein structure and proteomics): The identification, characterization and quantification of all proteins involved in a particular pathway, organelle, cell, tissue, organ or organism that can be studied in concert to provide accurate and comprehensive data about that system.

- **Computational Biology** (includes molecular modeling, tissue engineering, organ modeling and systems biology) The development and application of data-analytical and theoretical methods, mathematical modeling and computational simulation techniques to the study of biological systems.
- **Drug Discovery** (includes lead development, compound screening, and molecular docking) The discovery and development of compounds with the desired potency and selectivity, lack of toxicity and appropriate characteristics to enable it to reach its target in vivo, and then enter the early stages of development, where further large-scale investigations are undertaken.
- **Computer-aided diagnostic imaging and image-guided interventions** (includes segmentation, registration, and volume rendering as well as image guidance for surgical interventions) The development of imaging for image-guided therapy, as well as molecular, functional, cellular, and genetic imaging tools, combining new information technology and image fusion/integration capabilities.

## 1.1 Case Studies

We present a case study for a selection of the categories above. While each case study represents the needs identified by an individual researcher, we selected the case studies that we hope are representative of the entire field and reflect the diversity of computational needs.

Below each case study we also summarize the needs identified within each specialty. There is, however, not always a clear separation between research categories. Bioinformatic tools, for example, have impacted a broad range of research activities and are used by researchers in genomics and proteomics as well as by protein chemists and systems biologists. Researchers using molecular dynamics to study docking may consider themselves to be computational biologist, although we have classified them under protein biochemistry.

Each of the chapters that follow focuses on a recognized research direction of bio-computing. Within each chapter is a presentation of the major computing thrust within that direction, a discussion of the key algorithms used by researchers, and a case study. We end the report with a case study of the DARPA funded, Virtual Soldier Project. The Virtual Soldier Project, which is

focused on complex mathematical models and visualization capabilities, represents an opportunity to develop and evaluate tools designed for a High Productivity Computing Systems.

## **Chapter 2: Bioinformatics**

Bioinformatics is the development and application of computer methods for management, analysis, interpretation and prediction for molecular biology. It encompasses networking, databases, visualization techniques, search engine design, statistical techniques, modeling and simulation, AI and related pattern matching, and data mining. For this report, we include genomics IT, DNA sequencing, and microarray technologies in this category. Genomics IT is complex text searching of DNA sequences used in DNA sequence assembly and analysis. There are two tasks in computational genomics: sequencing and analysis. Sequencing requires putting together millions of pieces of short error-prone sequences. Analysis of DNA sequences requires finding the individual genes and other biological features (there are approximately 30,000 human genes, which comprise only 2% of the genome). Computer solutions to these problems include alignment algorithms, probabilistic techniques for sequence analysis and large systems built from these basic algorithms.

The first step in the biological hierarchy is a comprehensive genome-based analysis of the rapidly emerging genomic data. Use of new microarray-based technologies make possible high-throughput approaches that are rapidly generating terabytes of information, potentially providing fundamental insights into biological processes ranging from gene function to development, cancer, aging and pharmacology. Modern sequencers produce 1000 base pair reads/sec and operate full-time for days at a time; continual improvements in technology increase the throughput. Even partial understanding of the data can provide valuable research information. At the same time the huge quantities of data are overwhelming conventional methods of biological analysis.

Most genomics text searching algorithms are “embarrassingly parallel”; they can be deconstructed into a large number of independent searches with little message passing between

jobs. When the independent jobs are completed, the final results are assembled. Computation is integer based. The computer systems used for the computation can be either SMPs, clusters of single CPU systems, or SMP clusters.

There is significant research into new kinds of statistical models for predicting RNA structure. Hybrids of Hidden Markov Models (HMMs) and neural nets, dynamic Bayesian nets, factorial HMMs, Boltzmann trees and hidden Markov random fields are among the areas being explored.<sup>4</sup>

## **2.1 Biological Data Explosion**

Biological data is now estimated to be doubling every six months. The GenBank Database alone grew from 680,338 base pairs in 1982 to 22 billion base pairs in 2002 (compared to 13.5 base pairs as of August 2001<sup>5</sup>) and is now doubling in around 10 months. Currently the database grows by more than 11,000,000 bases per day (See Figure 2.1). GenBank is the NIH genetic sequence database, an annotated collection of all publicly available DNA sequences, and is part of the International Nucleotide Sequence Database Collaboration, which is comprised of the DNA DataBank of Japan (DDBJ), the European Molecular Biology Laboratory (EMBL), and GenBank at the National Center for Biotechnology Information. These three organizations exchange data daily. Each GenBank entry includes a concise description of the sequence, the scientific name and taxonomy of the source organism, and a table of features that identifies coding regions and other sites of biological significance, such as transcription units, sites of mutations or modifications, and repeats. Protein translations for coding regions are included in the feature table. Bibliographic references are included along with a link to the Medline unique identifier for all published sequences.

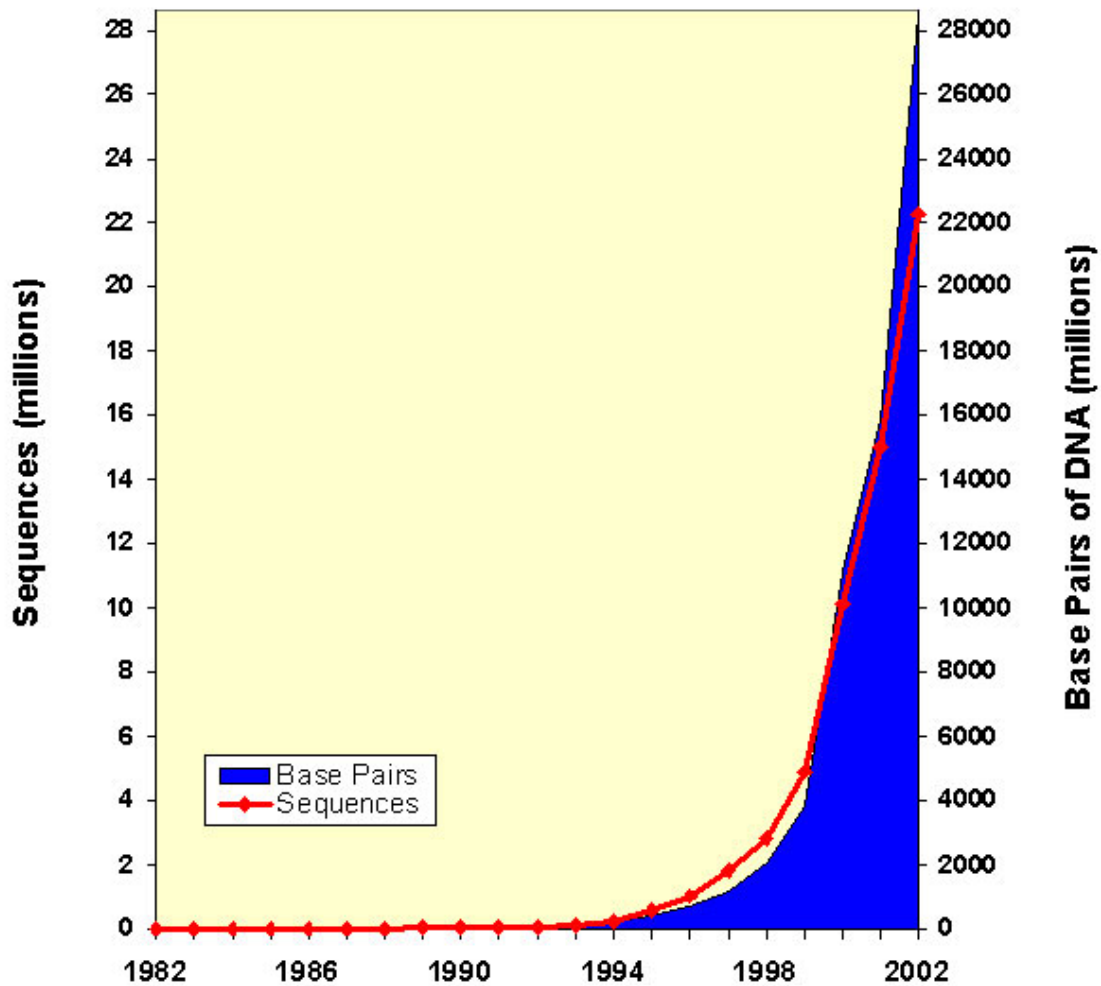


**Figure 2.1: Growth of GenBank**

Redundancies and database asynchrony are increasing because of the distributed and collaborative style of this research. As a result, database-to-database comparisons are required for analysis and validation, consuming ever more compute cycles and storage. The rate of acquisition of human and other genomic data over the next few years will be approximately 100 times higher than originally anticipated due to improved sequencing technology and methods. As complete genomes are sequenced, the length of DNA comparison strings will change from single genes to entire genomes, with a concomitant expansion in the time to compute. To look at

long-range patterns of expression synthetic regions on the order of 10's of megabases become reasonable lengths for consideration<sup>6</sup>. The challenges include: access, storage, and archiving.

## **2.2 Sequence Alignment Algorithms**

Biological R&D often requires the comparison of two or more sequences. Similarity comparisons evaluate the “closeness” of sequences to each other by computing a metric that includes a reward for allowed differences and penalty for disallowed differences. An objective function determines what rewards and penalties are important and how to combine these into the closeness metric. Corresponding to the modes of biology there are two types of sequence assessment: homology evaluation and contextual analysis. Both types of analyses and objective functions are used to determine the best alignment of the sequences in question.

Homology evaluation looks for evidence that biological sequences are related by evolution. Orthologs are related molecules that have been changed due to speciation, while paralogs are replicated molecules in the same organism that have been altered through generations of independent mutation. Homology analyses depend on a proper analysis of related sequences because it is necessary to predetermine notions of which mutations are allowed and the rate they can be expected to occur.

Contextual analyses are used to join together many small sequences into fewer, longer sequences. They can also be used to find vector contamination in sequences, evaluate primer candidates, and perform biochip design. Contextual analyses look for the common features among sequences without concern for whether the sequences have a common ancestor. The comparisons determine whether sequences overlap or are contained within another sequence<sup>7</sup>.

## **2.3 Pairwise Sequence Alignment Algorithms**

The objective of a sequence alignment algorithm is to position amino acid sequences so that the matched stretches of amino acids correspond to common structural or functional features. Gaps in the aligned sequences correspond to regions where polypeptide loops are deleted or inserted. Sequence alignment is a key component of many procedures for predicting the structure of a new

protein whose sequence has just been determined. There are three general types of sequence-alignment methods:

- Algorithms that attempt to match two sequences along their entire length
- Algorithms that search for local alignments involving sections (not necessarily continuous) from the sequences. The best known of these are Needle-Waterman and Smith and Waterman.
- Heuristic methods – BLAST and FASTA

There are cases where sequences share a similar region but are otherwise completely different. Take, for example, the amino acids in the active site of an enzyme or transcription factor binding sites in a DNA sequence. To handle these cases local multiple alignment algorithms have been developed. Usually they only look for ungapped alignments thereby avoiding the problem of choosing the optimal gap penalty. A discussion of some of the most popular sequence-alignment applications and algorithms follows.

### **Basic Local Alignment Search Tool (BLAST)**

BLAST is a general purpose similarity search tool that may be used in contextual and homology analyses. It has good sensitivity and very good specificity and can report multiple local alignments between sequences. The BLAST algorithm is a heuristic search method. The programs use the statistical methods of Karlin and Altschul<sup>8</sup>. For a detailed description of the BLAST algorithm see [http://www.blc.arizona.edu/courses/bioinformatics/book\\_pages/blast.html](http://www.blc.arizona.edu/courses/bioinformatics/book_pages/blast.html).

A public domain version of BLAST is available from the Blast server at

<http://www.ncbi.nlm.nih.gov/BLAST/>. There are many variants of BLAST, including:

1. **BLASTN** - Compares a DNA query to a DNA database. Searches both strands automatically. It is optimized for speed, rather than sensitivity.
2. **BLASTP** - Compares a protein query to a protein database.
3. **BLASTX** - Compares a DNA query to a protein database, by translating the query sequence in the 6 possible frames, and comparing each against the database (3 reading frames from each strand of the DNA) searching.

4. **TBLASTN** - Compares a protein query to a DNA database, in the 6 possible frames of the database.
5. **TBLASTX** - Compares the protein encoded in a DNA query to the protein encoded in a DNA database.
6. **BLAST2** - Also called *advanced BLAST*. It can perform gapped alignments.
7. **PSI-BLAST** - (Position Specific Iterated) Performs iterative database FastA

FastA compares a DNA sequence to a DNA database or a protein sequence to a protein database. Practically, FastA is a family of programs, which include: FastA, TFastA, Ssearch, etc.

<http://www2.ebi.ac.uk/fasta3/>. For a sketch of the algorithm see

<http://www.math.tau.ac.il/~rshamir/algmb/98/scribe/html/lec04/node14.html>.

### **Dynamic Programming and the Needleman-Wunsch Algorithm**

The Needleman-Wunsch Algorithm is widely used for aligning pairs of sequences. The algorithm finds the optimal alignment based upon the scoring matrix used. [Needleman and Wunsch, 1970]. The algorithm uses dynamic programming, which forms the basis for a number of widely used methods in bioinformatics. As mentioned in the introduction, sequence alignment is a 'hard' problem because there are an extremely large number of possible solutions, on the order of  $10^{30}$  for two sequences of length 100.

### **Smith-Waterman (many variants available)**

The Smith-Waterman algorithm finds optimal, local alignment of nucleotide or peptide sequences and is typically used when low to moderate sequence identity is expected.

Alignments are optimal because the algorithm considers all possible ways that two sequences can be matched up and reports the one with the best score. The Smith-Waterman algorithm is a database search algorithm based on the Needleman and Wunsch algorithm. The Smith-Waterman algorithm uses dynamic programming to take alignments of any length, at any location, in any sequence, and determines whether an optimal alignment can be found. Based on these calculations, scores or weights are assigned to each character-to-character comparison: positive for exact matches/substitutions, negative for insertions/deletions. In weight matrices, scores are added together and the highest scoring alignment is output. Smith-Waterman is superior to the

BLAST and FASTA algorithms because it searches a larger field of possibilities, making it more sensitive; however, individual, pair-wise comparisons between letters slow down the process significantly.

Instead of looking at an entire sequence at once, the Smith-Waterman algorithm compares multi-length segments, looking for whichever segment maximizes the scoring measure. The algorithm itself is recursive:<sup>9</sup>

$$H_{i,j} = \max\{H_{i-1,j-1} + s(a_i, b_j); H_{i-k,j} - W_k; H_{i,j-1} - W_1; 0\}.$$

Many groups use special hardware and software, such as Bioccelerator, to execute the algorithm. See [http://dapsas1.weizmann.ac.il/bcd/bcd\\_parent/bcd\\_bioccel/bioccel.html](http://dapsas1.weizmann.ac.il/bcd/bcd_parent/bcd_bioccel/bioccel.html).

### **Hidden Markov Models**

Hidden Markov Models (HMMs) are commonly used to specify protein profiles<sup>10</sup>. HMMs are built upon finite state machines with probabilities attached, i.e., stochastic regular grammars. HMMs have been generalized to recognize RNA secondary structure motifs using HMM algorithms with *stochastic context-free grammars* (SCFG) to capture conserved base-pairing. HMMs use position-specific scoring for the matching or substitution of a residue and for the opening or extension of a gap. HMMs are available from large, well-maintained libraries. HMMs have successfully been used in speech recognition as well as biology<sup>11</sup>. There are many variants available, see:

- HMMER is Sean Eddy's popular software for running HMMs. <http://hummer.wustl.edu>.
- SAM (Sequence Alignment and Modeling Systems) is HMM software developed by Richard Hughey, Kevin Karplus and David Haussler at UC Santa Cruz<sup>12</sup>.  
<http://cse.ucsc.edu/compbio/HMM-applicationsapps/HMM-.html>.
- HMMpro is commercial HMM software developed by Pieere Baldi and Yves Chavin at NetID Inc. (<http://www.netid.com>).

## tRNA Structure Modeling

Stochastic grammars can be applied to biological sequences. SCFGs, in particular, and the corresponding learning algorithms have been used to derive statistical models of tRNA. SCFGs, however, have some limitations. First, they are computationally intensive, so that in their present form they become somewhat impractical for long sequences, typically above  $N=200$ . Second, not all RNA structures can be captured by an SCFG. The associated parse trees cannot capture tertiary interactions such as pseudoknots and non-pairwise interactions. Third, they do not include a model for introns that present in some tRNA genes. Future requirements include<sup>13</sup>:

- Algorithmic and hardware speed improvements.
- Development of grammars, perhaps graph grammars, or other models, and the corresponding training algorithm to incorporate RNA tertiary structures, and possibly the tertiary structure of other molecules.
- Combination of SCFGs in modular ways, as for HMMs, to model more complex RNA sequences, including the corresponding introns.
- Modeling larger and more challenging RNA sequences, such as rRNA.
- Developing hybrid SCFG/NN architectures (or SG/NN), where NN is used to compute the parameters of a SCFG and/or to modulate or mix different SCFGs.

## 2.4 Multiple Alignment Methods

The Needleman and Wunsch algorithm for finding the best global alignment of two sequences can readily be extended to multiple sequences. The problem is that the time the computer needs for such a job is roughly proportional to the product of the sequence lengths. So, if aligning two sequences of 300 positions takes 1 second, aligning 3 sequences takes 300 seconds and aligning 10 sequences would take  $300 \times 10^8$  seconds, which is longer than the lifetime of the universe! Since searching for a best global alignment using a rigorous algorithm is not realistic for more than three sequences, a number of strategies have been developed to carry out a multiple global alignment in a reasonable amount of time with a reasonable chance of finding the best alignment.

### CLUSTAL W

CLUSTAL is one of the most popular packages for multiple sequence alignment. Multiple sequence alignment of nucleotide or protein sequences is an important tool in modern biology

that helps reveal similarities or differences between various sequences. Its main features include carrying out multiple alignments of a large number of sequences with additional features for profile alignments (alignments of old alignments) and phylogenetic analysis. (Neighbor Joining trees can be calculated after multiple alignment with a bootstrapping option). The CLUSTAL alignment algorithm consists of 3 steps:

- calculation of pairwise sequence similarities in order to calculate a distance matrix giving a divergence of each pair of sequences.
- construction of a guide tree (or a dendrogram) from the distance matrix.
- multiple alignment of the sequences in a pairwise manner according to the branching order in the guide tree.

### **FastME**

FastME is a fast phylogeny reconstruction program based on the minimum evolution method. Among distance methods, FastME has shown better topological accuracy than Neighbor Joining, BIONJ, WEIGHBOR and FITCH. FastME first builds an initial tree, using either GME or BME algorithms, and then improves this tree by tree swapping, using either FASTNNI or BNNI algorithms. GME and FASTNNI optimize the ordinary least-squares (OLS) version of the minimum-evolution principle, while BME and BNNI optimize the balanced version<sup>14</sup>. A public version is available at: <http://www.ncbi.nlm.nih.gov/CBBresearch/Desper/FastME.html>

### **PHYLIP**

PHYLIP (PHYLogeny Inference Package) is a package of programs for inferring phylogenies (evolutionary trees). It is maintained and developed by Dr. Joe Felsenstein ([University of Washington](http://www.washington.edu)). Methods that are available in the PHYLIP package include DNA and protein parsimony, distance matrix, and likelihood methods, including bootstrapping and consensus trees<sup>15</sup>.

### **Gibbs Sampler**

The Gibbs sampler algorithm involves iteratively making a profile with stretches of  $n$  bases or amino acids, selected from the sequences, and then searches this profile against one of the sequences. The result of the search is used to weight the selection of the stretches at the next run.

A drawback is that the user must choose the width  $n$  and the number of elements in each sequence and thus must have a certain idea of the outcome, or run the program several times. An interesting feature is that the Gibbs sampler algorithm avoids the choice of an externally added scoring scheme since it derives the highest scoring profile, in a self-consistent manner, from the data<sup>16</sup>. The Wadsworth Center of the New York State Department of Health maintains current versions of the Gibbs Motif Sampler at <http://bayesweb.wadsworth.org/gibbs/gibbs.html>.

## 2.5 Data Mining

With the dramatic increase in the amount of information stored in electronic format, the term 'Data Mining' (or 'Knowledge Discovery') has been coined to describe a variety of techniques to identify information or decision-making knowledge in bodies of data, and extracting this knowledge in such a way that it can be put to use in areas such as decision support, prediction, forecasting and estimation. Data mining techniques are an automated means of reducing the complexity of data in large bioinformatics databases and of discovering meaningful and useful patterns and relationships in data. Common data mining methods and tools are: feature selection, classification, and regression. We describe some of these methods and their applications in more detail below.

### Feature Selection

Feature extraction/selection is an important problem in knowledge discovery, not only for the insight gained from determining relevant modeling variables but also for the improved understandability, scalability, and possibly, accuracy of the resulting models. Feature extraction constructs a new set of variables by performing a linear or non-linear transformation on the old variables. Feature selection attempts to remove redundant features that do not provide additional information. A common way to classify feature extraction/selection algorithms is determined by how the learning method is integrated into the algorithm. A filter approach is where the selection of features is independent of the learning algorithm. On the other hand, if the features are generated and directly evaluated by a classifier or regression algorithm, the method is known as a wrapper approach.

Many feature selection algorithms can be classified into one of three groups: exhaustive search, heuristic search, and randomized search. Exhaustive search is a brute force approach where every possible subset is tested with the performance measure, and the best one is chosen. It guarantees the optimal subset as a result. If the number of features is large, however, this approach is intractable. Heuristic search is where a set of heuristics is used to greedily but intelligently search through the subset space to identify a subset with a reasonable performance measure. Forward Selection (FS) and Backward Elimination (BE)<sup>17</sup> are two examples of the heuristic search method. Both the FS and BE algorithms are iterative. FS starts with the empty set of features and proceeds to add additional features. Each additional feature is chosen to optimize the performance of the previous feature subset with the new feature added. The process stops when increasing the size of the current best subset leads to a lower performance. BE starts with the complete set of features and selects the as the next feature set the subset that optimizes the performance measure with one feature less than the current set. The process stops when decreasing the size of the current best subset leads to a lower performance.

There are many variants of Forward Selection and Backward Elimination. Randomized search uses randomized or probabilistic methods to search through the subset space. Genetic Algorithms<sup>18</sup> and Scatter Search algorithms<sup>19</sup> are examples of this approach. Both use the Darwinian evolution concept to progressively search for better subsets. Neither heuristic search nor randomized search techniques guarantee optimal results.

One area that has received much attention in the feature selection literature<sup>20 21 22</sup> is the identification of gene SNP patterns impacting cure or drug development for various diseases. Genomic studies provide large volumes of data with the number of single nucleotide polymorphisms (SNPs) ranging into thousands. The analysis of SNPs permits determining relationships between genotypic and phenotypic information as well as the identification of SNPs related to a disease.

### **Classification and Regression**

In statistics, classification is a type of statistical algorithm that takes a feature representation of an object or concept and learns to map it to a classification label. In a biological setting, the label

might refer to gene function, protein structure, or therapeutic outcome. Classification is as an instance of a broader class of machine learning algorithms. Two common algorithmic types in machine learning are supervised and unsupervised learning. Many of the tools used for classification are also used for regression problems.

Supervised learning is a machine learning technique for creating a function from training data. The training data consists of pairs of input objects and desired outputs. The output of the function can be a continuous value (see regression), or can predict a class label of the input object (called classification). The task of the supervised learner is to predict the value of the function for any valid input object after having seen only a small number of training examples. Unsupervised learning is a method where a model is fit to observations. It is distinguished from supervised learning by the fact that there is no *a priori* output.

Neural networks have become a popular tool for classification, as they are very flexible and do not assume any parametric form for distinguishing between categories. Applications can be found in both the frequentist and Bayesian literature. An aspect of neural network computational practices, which related to feature selection algorithms, is model selection. Much of the recent work on model selection using neural networks has been in the Bayesian framework; it includes Gaussian approximations for the posterior to approximate posterior probabilities<sup>23</sup> and reversible jump MCMC methods<sup>24</sup>. More established methods<sup>25</sup> include cross-validation and penalized likelihood methods using the Akaike Information Criterion (AIC)<sup>26</sup>, the Bayesian Information Criterion(BIC)<sup>27</sup>, or the Network Information Criterion (NIC)<sup>28</sup>. A Bayesian approach to model selection is Automatic Relevance Detection (ARD)<sup>29</sup>, which uses an additional layer of hyper-parameters to try to shrink unimportant variables. ARD, however, does not allow one to compute posterior probabilities of individual models.

An example of the propagation of bioinformatic tools into the medical field is the growth of nursing databases. These databases, which are known to be massive and multidimensional, easily exceed the capabilities of both human cognition and traditional analytical approaches. An emerging, innovative approach<sup>30 31</sup> to knowledge discovery in large databases takes advantage of Bayesian confidence propagation neural networks (BCPNN), a state-of-the art representation of

probabilistic knowledge by a graphical diagram. Bayesian networks allow investigators to combine domain knowledge with statistical data, enabling nurse researchers to incorporate clinical and theoretical knowledge into the process of knowledge discovery in large datasets.

## **2.6 Bioinformatics Case Study: Brian Athey, Ph.D.**

Dr. Athey is the Director of the Michigan Center for Biological Information (MCBI) at the University of Michigan. MCBI provides advanced bioinformatic and computational resources for investigators in the academic and industrial sectors of Michigan. Researchers have access to bioinformatics tools, genomics and proteomic databases, supercomputing resources, bioinformatics training, and bioinformatics consulting through MCBI. MCBI is researching appropriate hardware, middleware, and networking structures for statewide analysis and data-sharing in bioinformatics projects.

As director of MCBI, Dr. Athey is responsible for ensuring researchers using MCBI facilities have access to the best resources available. In this capacity, he has identified three problems that need to be addressed:

1. Data glut: As data sizes grow data motion will bottleneck the computing progress. For example, in the 100 seconds it takes to move the 1 GB file, a 1 GHz machine could perform 0.1TeraOP. Data needs to be local, and stay local. Data motion needs to be asynchronous, and happen at near wire speeds. Throwing money at the network does not solve the problem.
2. Data storage: Many PC file systems hit the limits in the TB range. This will be problematic in the next year or two. A 1 TB database would require (ignoring other issues), 20000 seconds to read at 50 MB/sec.
3. Compute cycles: Most bioinformatic applications are: integer bound; memory latency bound, and pointer chasers (cache thrashers). IA32 machines offer the best price performance for these classes of computations.

Dr. Athey shares the feeling that general needs in biomedical sciences can be enabled by next generation supercomputing. Projects that will be enabled by HPSCS include mouse/human

genome correlation, individual pharmacogenomic analysis using gene expression arrays, multi-modal radiology image fusion, millisecond structural biology enabled by synchrotron x-ray sources and 900 MHz NMR, and physiologically competent Digital Human Simulations. He notes that not all biology problems are embarrassingly parallel; shared memory with database(s) close in is preferred in many (most) biologically interesting problems.

The World Wide Web and bioinformatic databases have had a profound affect on the research activities of biologist. Bioinformatic tools have put the information available in these resources at the fingertips of researchers. Dr. Athey has identified several key informational needs of researchers. He anticipates the need for an almanac or index that would link every human gene to all the information known about these genes from the literature, from all relevant expenditures and other sources. In addition, better relational databases would help researchers to move from a gene by gene approach to focus more on patterns and pattern recognition. Better and system wide in silico models of human would allow researchers to begin to be able to understand how proteins are modified in disease states and obtain more detailed information on the structures of drug targets.

## **2.7 Bioinformatics Needs**

Some thought has been given as to how to present the results of our informal survey. Due to the limited sample size, it would be inappropriate to represent the needs of the individuals selected for interviews as exhaustive of those of the entire spectrum researchers in the field in question, whether it is in bioinformatics, computational biology, protein biochemistry, or visualization. We can hope, however, that while our survey does not capture every conceivable need and requirement, it does capture a large subset of the core needs of the fields under consideration. With this in mind, we present a list of needs gathered from the interviewees who were willing to participate in our study. We would like to thank them all for their input. While many researchers interviewed were involved in projects spanning our categories, we grouped their comments according to which of our classification of bio-computing areas the comments appeared to fit in most appropriately.

## **Interviewees for Bioinformatics**

### **Brett Peterson, Ph.D.**

Dr. Peterson is health scientist administrator in National Center for Research Resources (NCRR's) Division of Biomedical Technology. The interview was held at the January 2003 workshop. He discussed the Biomedical Informatics Research Network (BIRN), a [National Institutes of Health](#) initiative that fosters distributed collaborations in biomedical science by utilizing information technology innovations.

### **Brian Athey, Ph.D.**

Brian Athey is Director of the Michigan Center for Biological Information at the University of Michigan. MCBI provides advanced bioinformatics and computational resources for investigators in the academic and industrial sectors of Michigan.

### **Stanley K. Burt, Ph.D.**

Stan Burt is the Director of the National Cancer Institute's Advanced Biomedical Computing Center (ABCC). The National Cancer Institute's supercomputing facility is a fully integrated, high performance, scientific computing resource located at the [NCI-Frederick](#) campus in Frederick, MD.

### **Ron Elber, Ph.D.**

Ron Elber is a Professor in the Department of Computer Science at Cornell University. He is also on the faculty of the Cornell Genomics Initiative, Computational and Statistical Genomics Focus Area. He is active in two areas of research: bioinformatics and molecular dynamics.

## **Requirements**

1. There is a need to establish distributed and linked data collections for investigators' research projects; enable access to heterogeneous "grid-based" computing resources for research project analyses; provide data mining tools to search multiple data collections or databases; develop the software and hardware infrastructure that will allow scientists to conduct valid multi-site, neuro-imaging studies, for example.

2. Infrastructure must permit research that focuses on combining data from multiple acquisition sites and increasing the statistical power for studying relatively rare populations.
3. Researchers need to operate in a heterogeneous computer environment in order to let them match their specific problem needs to the appropriate platform.
4. As data sizes grow data motion will bottleneck the computing progress. Data needs to be local, and stay local. Data motion needs to be asynchronous, and happen at near wire speeds. Throwing money at the network does not solve the problem.
5. Data size issues dominate processing time, data motion and data storage. Data must be distributed, Data I/O must occur over many channels, and have no single points of flow.
6. Bioinformatic applications depend on access to large quantities of data and often load gigabytes of data into memory at once. In addition, in a distributed environment, the data must be shared across all nodes and each node must be capable of storing at least 30 gigabytes.

Bioinformatics is a rapidly growing area of research. Most biologists talk about "doing bioinformatics" when they use computers to store, retrieve, analyze or predict the composition or the structure of biomolecules. While bioinformatics "grew up" dealing primarily with sequence analysis, bioinformatics tools have been incorporated into diverse fields, including comparative genomics, medical informatics, computational biology, cheminformatics, genomics, mathematical biology, proteomics, pharmacogenomics, and pharmacogenetics. The bioinformatics research community stands to benefit greatly from advances in high-productivity computing that enable seamless data-sharing and increased computational speed.

## **Chapter 3: Computational Protein Biochemistry**

Determining the shape of proteins from their sequences is one of today's great computational challenges. The properties of any protein are largely determined by its structure. Proteins usually adopt a single structure, corresponding to the global minimum free energy under physiological conditions. Protein sequences can generally fold into a unique state in just a few seconds (or

less) from any starting conformation. Protein structures can be experimentally determined by crystallizing the protein and then using x-ray crystallography or NMR to find the position of the atoms, but this is a difficult procedure. The experimental process of deciphering the atomic structures of the majority of cellular proteins is expected to take a century at the present rate of work. Thus, there is strong interest in using computational methods to predict protein structure. The folded structure of a sequence is determined by the sequence of successive solid bend angles, where each angle can be represented by two planar angles. It is possible to make such a problem discrete by limiting the ways to bend each angle, but doing so decreases the accuracy of the solution. Even with such techniques, a 100-residue protein would have a search space of  $7^{100}$  ( $\sim 10^{84}$  configurations).

The Levinthal Paradox describes the discrepancy between the time for an exhaustive search of all possible conformations and the observed timescale of protein folding. If it is assumed that there are three conformations for each amino acid then a polypeptide chain with say 1—amino acids would have about  $10^{48}$  conformations. If the interconversion between conformations required just  $10^{-11}$  seconds then it would take about  $10^{29}$  years to explore them all. Of course, this is for the most basic of grid search algorithms, but even the most advanced systematic conformational search would still require an inordinate amount of time to identify the global minimum energy conformation.

Example: (Duan and Kollman 1998). 1 us simulation of a 36-residue peptide starting from a fully extended state. This peptide is one of the smallest proteins that can fold autonomously, with folding estimated to take between 10 us and 100 us. It contains three short alpha-helices. The simulation involved in addition to the protein about 3000 water molecules and was performed in a truncated octahedron simulation box with a time step of 2 fs. About 4 months of computing time on a 256-processor parallel computer was required for the 1 us simulation. While the protein did not actually fold into the known experimental structure, a marginally stable state which showed significant resemblance to the native conformation was observed. This state had a lifetime of about 150ns.

Protein Biochemistry includes protein structure and proteomics. It includes the identification, characterization and quantification of all proteins involved in a particular pathway, organelle,

cell, tissue, organ or organism that can be studied in concert to provide accurate and comprehensive data about that system.

### 3.1 Protein Folding

A variety of approaches have been used for protein folding. The most ambitious approaches attempt to solve it *ab initio*. The conformational space of the molecule is explored to identify the appropriate structure. The total number of conformations is very large, and so it is usual to try to find only the very lowest energy structures. Some form of empirical force field is generally used, often augmented with a solvation term. The global minimum in the energy function is assumed to correspond to the naturally occurring structure of the molecule.

Rule-based methods, often called threading, have also been used for protein folding. This approach first determines which stretches of amino acids should adopt each type of secondary structure and then packs these secondary structural elements together to achieve a low-energy structure. The threading approach relies on the quality of the initial secondary structure prediction. It works best if the structural class to which the protein belongs is known. A third approach, comparative modeling, exploits the structural similarities between proteins by constructing a 3-D structure based on the known structure(s) of one or more related proteins. When using comparative modeling, one must initially determine which protein structure(s) to use as the 3D templates, and then decide how to match the amino acids in the unknown structure with the amino acids in the known structure(s). Each of these methods is described in the following sections.

#### ***Ab Initio* Prediction**

*Ab initio* prediction programs work by defining a global energy function and performing a search of possible bond-angle configurations to find one which minimizes total energy. *Ab initio* approaches explore the conformational space of the molecule to identify the appropriate structure. Since the total number of possible conformations is very large, it is usual to try to find only the lowest energy structures. Some form of empirical force field is usually used, often augmented with a solvation term.

Many methods are used for exploring the conformational space, many of which are analogous to the models used to perform Monte Carlo simulation of polymers, such as the lattice and ‘bead’ models. An optimization procedure based on simulated annealing or a genetic algorithm is often used with simplified molecular dynamics models to first identify families of low-energy structures, which may then be converted into a more detailed representation for subsequent refinement. The most important issues are:

- 1) the energy function selected – energy minimization functions include hydrophobic/hydrophilic interactions; size and flexibility properties of different amino acids; and electrostatic/Van der Waals interactions of nearby atoms;
- 2) the optimization procedure employed to search the space – methods include gradient descent, simulated annealing, and genetic algorithms, possibly using parallel computation.

The General Atomic and Molecular Electronic Structure System (GAMESS) is a general *ab initio* quantum chemistry package. GAMESS is maintained by the members of the Gordon research group at Iowa State University. For more information, visit:  
<http://www.msg.ameslab.gov/GAMESS/GAMESS.html>.

### **Threading or Fold Assignment Approaches**

Many programs use known 3D structures to help determine a protein’s 3D structure. Two amino acid sequences with 20% - 30% identical residues likely have similar 3D structures. Threading, or inverse folding, programs are commonly used. The basic concept is to choose from among a number of 3D protein structures, typically chosen to represent a common structural class, chose the structure most compatible with the sequence of the unknown protein. This is accomplished by “threading” the sequence through each protein structure in turn. Threading methods are closely related to *ad initio* approaches to protein structure prediction, but threading methods inherently limit the search space to the conformations of known structures.

Threading programs use special searching methods such as double dynamic programming to efficiently find the best ways to match the sequence to the structure. Approximations are used to make the problem more manageable. Many of the scoring functions used in threading algorithms are potentials of mean force that provide an estimate of the free energy of interaction between two residues as a function of their separation. These potentials of mean force are calculated from statistical analyses of known protein structures. For threading algorithms one is particularly interested in the interactions between amino acids that are close in 3D space but far apart in the sequence, and the potentials used in such calculations are derived appropriately. In addition, the pairwise knowledge-based term, a solvation contribution, is often added. Although knowledge-based potentials are most popular, it is also possible to use other types of potential function.

No one single theoretical or experimental technique can predict protein function from sequence, rather it is the application of an appropriate combination of methods that is required. There are two main approaches:

- 1) developing potentials for fold assignment
- 2) HMMs that are descended from alignment methods

The input is 1) a protein structure, 2) a core model describing the position of the core residues and allowable lengths of loops and 3) a scoring function to evaluate the given threading. Without modeling pairwise interactions this is a simple dynamic programming problem. It has been estimated that the success rate of fold assignment algorithms will increase to roughly 50% once the library of protein folds grows. For the remaining genome sequences to be assigned to folds, it will be necessary to move to multi-positional compatibility functions. Incorporating pairwise interactions will require tabulating the possible substructures for every base assignment, not just the best matching prefix structures.

HMMs are used for fold identification by performing a standard sequence-based homology search using the probe sequence to generate homologous sequences. These sequences can be used to construct an HMM based on the probe, and then sequences from a library of folds can be

matched against the HMM. HMMs can also be used to construct separate HMMs for each member of a library of folds and then score the probe sequence against each model. Construction of HMMs is typically an iterative process involving successive periods of model building, searching with the given model, and model refinement. Alignment to an HMM can be performed in an efficient recursive manner, similar to dynamic programming.

### **Comparative Modeling**

Comparative modeling exploits the structural similarities between proteins by constructing a 3D structure based upon the known structures of one or more related proteins. To do this, it is necessary to decide which protein structures to use as 3D templates, and then to decide how to match the amino acids in the unknown structure with the amino acids in the known structures. Comparative modeling methods consist of the following sequential steps:

- 1) identify the proteins with known 3D structures that are related to the target sequence
- 2) align these with the target sequence and pick those known structures that will be used as templates
- 3) build the model for the target sequence given its alignment with the template structures
- 4) evaluate the model against selected criteria
- 5) if necessary, repeat the alignment and model building until a satisfactory evaluation is reached.

In a typical comparative modeling exercise one would use a heuristic algorithm to determine possible sequences of interest, then the Smith-Waterman method to identify the appropriate subsequences, and finally the Needleman-Wunsch algorithm to derive the alignment to use in the actual construction of the model.

There are three different classes of method for constructing the 3D model. Generally, each of these three methods is used, with construction proceeding as in the following three-stage process.

1. Piece together rigid bodies taken from the template protein(s). This step constructs the model from a few core regions, loops and side chains obtained from dissected related structures.
2. Assemble the target protein by joining together small segments or by reconstructing a set of coordinates. Segment matching relies on approximate positions of conserved atoms from the templates to calculate the coordinates of other atoms; this is achieved by the use of a database of short segments or protein structure, energy or geometry rules, or some combination of these criteria. Fragment assembly without using an underlying framework. The fragments are taken from proteins of known structure which show local sequence similarity to the unknown target. The initial structures resulting from this “splicing” process are then subjected to simulated annealing using a scoring function that has sequence-dependent terms and sequence-independent terms. The most promising of several runs are selected.
3. Generate a series of spatial restraints from the templates, which are used in conjunction with an optimization procedure to derive a structure of the target. Satisfaction of spatial constraints uses either distance geometry or optimization techniques to satisfy spatial restraints obtained from the alignment of the target sequence with homologous templates of known structure. The optimization uses a combination of conjugate gradients, with molecular dynamics and simulated annealing.

### **3.2 Computational Protein Biochemistry Case Study: Ron Elber, Ph.D.**

Dr. Ron Elber is a Professor in the Department of Computer Science at Cornell University. He is also on the faculty of the Cornell Genomics Initiative, Computational and Statistical Genomics Focus Area. He is active in two core areas of research: bioinformatics and molecular dynamics. In bioinformatics, he is interested in protein annotation (structure and function prediction), protein evolution, protein folding potentials, and protein alignment. In protein dynamics, he develops theory, algorithms, and computer code to simulate bio-molecular dynamics, the long dynamics of biophysical processes, and protein folding. Among the substances that have been studied in detail by Dr. Elber are the oxygen transport proteins hemoglobin and myoglobin and ion channels such as gramicidin.

Dr. Elber notes that bioinformatics and molecular dynamics present different computational demands. A molecular dynamic simulation of protein folding with a medium size protein of 150 amino acids and from 1000 to 100000 particles can run for a month on a cluster of 100 off-the-shelf CPU's. On the other hand, most bioinformatics applications are much more rapid, completing within minutes or hours. While molecular dynamic simulations tend to require raw processing power, bioinformatic computation places a premium on memory and data sharing. These differences in computational needs mean that compromises must be met when purchasing new hardware.

In addition, the different computational needs between bioinformatics and molecular dynamic applications can also be seen at the memory usage and I/O performance levels. Bioinformatic applications depend on access to large quantities of data and often load gigabytes of data into memory at once. In addition, in a distributed environment, the data must be shared across all nodes and each node must be capable of storing at least 30 gigabytes. Molecular dynamic applications have limited I/O and memory demands, only requiring on the order of 100 megabytes.

Dr. Elber identified several sites where there is room for improvement.

1. Fault tolerance in a distributed environment: Although local, in house fixes have adequately met current needs, O/S level changes would be appropriate.
2. Platform porting: With special care, code can be made portable, especially across alternative unix/linux flavors. Windows, however, presents additional challenges, especially with stability.
3. Code management: Current code management tools have been found to be too restrictive in an academic setting and are convenient to use. Current applications have  $10^5$ - $10^6$  lines of code.
4. Debugging tools: Productivity would improve with better debugging tools.
5. Algorithmic: Some machine learning tools manipulate matrices with  $10^{16}$  elements. There is little work being done to develop algorithms to manipulate very large data sets in a distributed environment.

A current growing concern in the biomedical community is the need to develop tools and theories dealing with the multiple temporal and spatial scales. One of the striking observations in dynamics of biological molecules is the extremely large time scale they covered. Initiation by light absorption of biochemical processes is very rapid (femtoseconds), while protein folding is slow (milliseconds to minutes). Current molecular dynamic approaches are restricted to nanoseconds ( $10^{-9}$  seconds). Multi-scale modeling must maintain the detail description at the molecular level but be capable of generating a description of macro-level biology. Even if computer performance increases by a factor of two each year, this will be outpaced by the tremendous advantages that can be obtained by working on theory and algorithms, which are capable increasing performance by a factor of millions.

With an eye toward the future, Dr. Elber notes that current directions in bioinformatics will soon require that very large databases stored at multiple sites are able to be accessed, placing large demands on I/O and memory. Stability of systems, especially windows-based, will become a larger issue. Within five years, he would like to be able to access databases that are a factor of 1000 times larger than currently in use. It will then be possible to begin to answer more challenging questions about the nature of the interaction between genetic changes at the molecular level and the environment. Researchers will be able to correlate protein structure and genomic information with the different observed phenotypes. This will allow us to gain a better understanding of the interaction among species and life on earth.

### **3.3 Computational Protein Biochemistry Needs**

#### **Interviewees**

##### **John Yate, III, Ph.D.**

John Yates is a Professor of Cell Biology at the Scripps Research Institute, where he is director of the Proteomics Mass Spectrometry Lab. Tandem mass spectrometry is a powerful technique for characterizing a proteome. Proteomics by tandem mass spectrometry requires powerful informatics capabilities.

**Giri Chukkapalli, Ph.D.**

Dr. Giri Chukkapalli received his PH.D in Mechanical Engineering at the University of Toronto, focusing his dissertation on developing weather models on parallel computers. He is an assistant programmer/analyst at SDSC, where he is involved with several projects, including code (MPI) parallelization, the IBM S/390 supercomputer, and research involving computational fluid dynamics.

**Ron Elber, Ph.D, Stanley Burt, Ph.D. (see Chapter 2: Bioinformatics)****Requirements**

1. Hardware/software are needed to support the pipeline processing efficiently (other fields have similar needs, including climate modeling). Tools would include scheduling and checkpointing. Well-balanced hardware pipeline from archival storage to compute elements without bottlenecks and easily programmable FPGA coprocessor boards to handle integer and other DSP branch of the pipeline are needed. Hardware and software to handle the overlapped computation, communication and I/O would improve efficiencies.
2. Efficient ANN and GA libraries similar to LAPACK would assist in code development and improve time to solve.
3. To drive down the cost of I/O operations, copies of sequence databases are stored locally. The growth in size of sequence databases will eventually stress memory capabilities.
4. The current algorithmic bottleneck in mass spec. analysis occurs in the initial pass through the sequence database to identify amino acid sequences that match the measured mass of peptides under consideration. More efficient search algorithms could increase productivity by a factor of 10 to 100.
5. Space and cooling are significant cost factors for Beowulf clusters.
6. Collaboration with other research institutes could be facilitated with higher bandwidth internet communications. Experience has shown that it is often faster to run an analysis on a mass spectrometer dataset on slower hardware than it is to ftp that same dataset to another site.

Computational protein biochemistry has become an important area of scientific research. Computational methods, such as Molecular dynamics simulation methods, continuum electrostatics and a variety of empirical solvation models, which were developed to solve problems in this field, have become widely used tools in physical biochemistry and structural biology. While enormous progress that has been made, a number of challenges have been elusive, for example the inability of existing methods to consistently predict the relative binding free energies of different substrates to the same protein or the conformation of loops which connect two fixed secondary structure elements. In the absence of a well-defined physical model, database mining suggests opportunities for computational biochemistry; statistical methods resulting from database mining of one type or another are the most of the successful in actual fold prediction to date. High productivity computing systems will hasten the day when the process of multiple sequence analysis, structural prediction, the design of combinatorial libraries and binding free energy calculations will be carried out in a single group by researchers who understand the intricacies of each of these problems.<sup>32</sup>

## **Chapter 4: Computational Biology**

Computational Biology includes molecular modeling, tissue engineering, organ modeling and systems biology. It is the development and application of data-analytical and theoretical methods, mathematical modeling and computational simulation techniques to the study of biological, behavioral and social systems.

*Computational biology techniques can play an important role in countering bioterrorism. Since these methods are used to determine structure-function relationships in proteins in order to understand the biological pathways, it also provides the tools to study the factors that lead to toxin formation and the interruption of such pathways, either for detection, prevention, remediation or health impacts. Computational approaches can be used for the rational redesign of enzymes to degrade chemical agents. An example is the enzyme phosphotriesterase (PTE), which could be used to degrade nerve gases. Combined experimental and computational efforts can be used to develop a series of highly specific PTE analogues, redesigned for optimum activity at specific temperatures or for optimum stability and activity in non-aqueous and low humidity environments or in foams, for improved degradation of warfare neurotoxins. It is also possible to use advanced computations to design more efficient therapeutic agents against the highly toxic phosphoester compounds such as the nerve warfare agents DFP, sarin, and soman and insecticides like paraoxon - **Grand Challenges in Computational Structural and Systems Biology** (DA Dixon, TP Straatsma, T. Head-Gordon – PNNL & LBNL)*

## 4.1 Molecular Modeling

Molecular modeling includes biochemical analysis, protein binding/drug target evaluation, and dynamics of molecules. Molecular modeling techniques are widely used in the chemical, pharmaceutical and agrochemical industries. Molecular modeling techniques allow the simulation of systems of variable size, ranging from a few tens to millions of atoms. The parameters which rule the reliability of the simulation reside on the accuracy in the definition of the inter-atomic potential and on the dimensions of the investigated system, since a higher accuracy usually corresponds to increased computational requirement which in turn limits the dimension of the system under study.

A number of popular software packages for molecular modeling are available and used widely, including:

- GAMESS programs for *Ab initio* Quantum Chemistry
- GAUSSIAN programs for *Ab initio* Quantum Chemistry [www.gaussian.com](http://www.gaussian.com)

- NWCHEM programs for Quantum Mechanics
- CHARMM programs for molecular mechanics <http://yuri.Harvard.edu>.
- AMBER programs for molecular mechanical force field <http://amber.ucsf.edu>.
- MOPAC/AMPAC programs for semi-empirical quantum mechanics.
- MM2 program for molecular mechanics.

A variety of modeling techniques have been developed over the years including:

- Quantum Mechanical Methods.
- Energy minimization.
- Molecular dynamics-based techniques involving path-integral and Monte Carlo methods.
- Molecular dynamics combined with electron density function theory.
- Conformational analysis.
- Cellular automata.
- Lattice Boltzmann method<sup>33</sup>.

In the following paragraphs, we present these modeling techniques in more detail.

### **Quantum Mechanical Method**

Quantum mechanical (QM) methods are used to determine energy interaction potentials. QM methods deal with the electrons in a system, so that large numbers of particles must be considered and the calculations are time-consuming. Quantum mechanics explicitly represents the electrons in a calculation, and so it is possible to derive properties that depend upon the electronic distribution and to investigate chemical reaction in which bonds are broken and formed. There are two major categories of quantum mechanical molecular orbital calculations: *ab initio* and semi-empirical methods. The *ab initio* method uses the full Hartree-Fock/Roothaan-Hall equations, without ignoring or approximating any of the integrals or any of the terms in the Hamiltonian. Semi-empirical methods simplify the calculations, using parameters for some of the integrals and/or ignoring some of the terms in the Hamiltonian. Many different programs are available for performing *ab initio* calculations, the best known of these is the Gaussian series of programs.

An ab initio calculation [35, 36, 42, 48, 33] can be logically considered to involve two separate stages. First, the one- and two-integrals are calculated. This is computationally intensive. In the second stage, the wavefunction is determined using the variation theorem. In a traditional Self-consistent Field (SCF) calculation all of the integrals are first calculated and stored on disk, to be retrieved later during the SCF calculation as required. The number of integrals to be stored may run into millions and this leads to delays in accessing the data. In direct SCF calculation, the integrals are not stored on the disk but are kept in memory or recalculated when required<sup>34</sup>

Ab initio methods represent the higher level of description of the inter-atomic potential and allow, in principle, the exact solution of the Schrödinger equation without the introduction of any parameters. However, they usually offer a particularly unfavorable scaling with the dimensions of the system,  $O(M^8)$ , typical of many-body problems, which makes them applicable to systems composed by a limited number of atoms, usually of the order of 10–20. However, they can be extremely accurate, up to 0.5 kcal/mol, and are still successfully applied in the field of atmospheric chemistry and elementary chemical processes, in which high accuracies are needed. From a computational point of view, the most intensive tasks are represented by the analytic or numerical evaluation of 2-electron integrals, and integral transformations from an atomic orbital to a molecular orbital base. Conventional algorithms require the storage on disk of an enormous amount of data, the semi-transformed integrals, of the order of tens of Gbytes, as well as large memory storage, bandwidth and latency. This class of applications can be defined as memory-bound and is indeed bound to the efficiency of the memory access on a single processor; moreover, the extremely involved data connectivity, makes these algorithms difficult to implement on parallel machines requiring more and more efficient single CPUs.

### **Energy Minimization**

Energy minimization is widely used in molecular modeling and is an integral part of techniques such as conformational search procedures. Minimum energy arrangements of the atoms correspond to stable states of a system. Energy minimization is also used to prepare a system for other types of calculations, for example it may be used prior to a Molecular Dynamic (MD) or Monte Carlo simulation. Molecular mechanics minimizations are nearly always performed in

Cartesian coordinates where the energy is a function of  $3N$  variables. Minima are located using numerical methods which gradually change the coordinates to produce configurations with lower and lower energies until the minimum is obtained. Algorithms used include the simplex or steepest descents methods and the Newton-Raphson algorithm. Systems containing thousands of atoms can require significant memory, and are usually solved using molecular mechanics methods.

### **Force Field Methods**

Force field methods, also known as molecular mechanics, are used to perform calculations on systems containing significant numbers of atoms. Force field methods ignore the electron motions (the focus of quantum mechanics) and calculate the energy of a system as a function of the nuclear positions only. In some cases force fields can provide answers that are as accurate as even the highest-level quantum mechanical fraction of the computer time. Molecular mechanics is based on a simple model of the interactions within a system with contributions from processes such as the stretching of bonds, the opening and closing of angles and the rotations about single bonds.

The interaction potential is usually expressed as the sum of apriori parameterized van der Waals and Coulombic contributions, the latter showing a quadratic scaling,  $O(M^2)$ , with the dimensions of the system (generally indicated with  $M$ ), due to the double sum on the atomic effective charges. The maximum accuracy of this class of methods is typically of the order of 20 kcal/mol, usually too limited for the description of phenomena of chemical interest. However, FF-based methods, implemented according to the fast-multipole (FM) expansion of the Coulomb potential, have been applied to the approximate description of systems containing up to a few million of atoms and have found a wide success in the investigation of biological systems, surface science and material science (e.g. protein science, material fractures, liquid crystals). FF-based algorithms usually offer excellent scaling performances on massively parallel architectures. FM expansion of the Coulomb potential can be implemented in such a way as to reach a linear scaling with the dimensions of the system [22].

## Density Functional Theory

Density Functional Theory (DFT) only attempts to calculate the total electronic energy and the overall electronic density distribution. DFT methods [39, 31, 19] usually offer an  $O(M^3)$  scaling with the dimensions of the system, typical of direct diagonalisation techniques. More favorable scaling, of the order of  $M^2 \log M$ , can be achieved by using a plane wave (PW) expansion as a basis set and pseudo-potentials (PP) for the description of core electrons [28]. The typical accuracy of these methods, of the order of 3-7 kcal/mol, makes them suitable to the study of chemical interesting problems, and DFT methods have been successfully applied to the investigation of chemical reactivity and complex material in systems composed by up to a few hundreds of atoms. Moreover, recent developments [24], include coupling the evaluation of a DFT potential to a classical molecular dynamics (MD) scheme, introducing time as a further degree of freedom to explore. The basic algorithmic features of DFT-based MD methods reside in the use of efficient fast Fourier transform (FFT) techniques to compute the different contributions to the total energy (kinetic energy, Coulomb, XC, PP) and its derivatives, the latter task being particularly computationally intensive. The large number of PWs, typically of the order of  $10^5$ - $10^6$ , necessarily translates in large memory requirements; moreover, the parallel implementation of this class of algorithms requires an extremely efficient communication network, due to the particular implementation of the parallel FFT which requires global data exchange. To give a measure of the memory and CPU requirements, a system composed by 350 atoms can require up to 24 Gbyte of memory and, to be executed in a reasonable time, 32 IBM power 3 processors. This class of methods will therefore benefit from both increased computer power, communication and memory bandwidth, even if it will be probably limited to run on proprietary hardware, due to the reduced performances of COTS communication devices. Moreover, an adequate development of scientific libraries (FFT and linear algebra) is needed to retain high performance for this class of algorithms.

Within the next years, DFT methods will probably allow the accurate computation of electronic, structural and dynamical reactive properties of systems containing 3000 (10000) atoms. Based on this, DFT methods will substitute FF parameterizations in chemiometric applications, in which a large number of medium-size calculations is needed. This will have a direct impact in

pharmacology: “the design of a new drug usually requires a pre-selection operated by computer simulations and data analysis; the advantage of a much higher accuracy in the description of the investigated molecular systems and properties directly translates into a high selectivity of the target system with a significant reduction of the number of laboratory tests, up to a factor of 10. DFT methods will also allow the accurate simulation of small protein systems, or of realistic portions of them, with particular impact on the comprehension of the action mechanism of metalloenzymes, where a reduced model usually neglects the fundamental underlying interactions. To understand the importance of such a field, it is sufficient to mention that both respiration and photosynthesis involve metallo-organic active centres constituted by several thousand atoms; comprehension of the action mechanism of such systems will allow to device efficient synthetic bio-mimetic analogues of the natural systems, with a high impact in the field of energy storage and molecular sensors. Moreover, we can predict that DFT-based methods will allow the accurate simulation of nano-scale systems with a high impact in the design of molecular engines, quantum computation devices and chemical storage of data [18].”

### **Conformational Analysis**

The physical, chemical and biological properties of a molecule depend upon the 3D structures or conformations that it can adopt. Conformational analysis is the study of the conformations of a molecule and their influence on its properties.<sup>35</sup> A key component of a conformational analysis is the conformational search, which seeks to identify the preferred conformations of a molecule; this requires locating conformations that are minimum points on the energy surface. The conformational search is concerned only with locating minimum energy structures.

### **Evolutionary Algorithms and Simulated Annealing**

Evolutionary algorithms and simulated annealing have found widespread use in molecular modeling, including use in finding the global minimum energy conformation of a molecule, protein-ligand docking, molecular design, Quantitative Structure-Activity Relationships (QSAR) and pharmacophore mapping<sup>36</sup>.

Evolutionary algorithms (EA) are a group of methods based on ideas of biological evolution that are designed to find optimal solutions to problems. There are three basic classes of evolutionary algorithm:

Genetic algorithms (GA)

Evolutionary programming (EP), and

Evolutionary strategies (ES).

All three are based on the concept of creating a 'population' of possible solutions to the problem. The members of the population are scored using a 'fitness function' that measures how 'good' they are. The population changes over time and evolves towards better solutions.

### **Genetic and Evolutionary Algorithms**

The main difference between the genetic algorithm and evolutionary programming is that the latter does not use a crossover operator. Evolutionary strategies are very similar to evolutionary programming but differ in two key respects: crossover operators are permitted and the probabilistic tournament is replaced with a straightforward ranking.

Genetic and evolutionary algorithms involve a significant random element and so they are not guaranteed to produce the same global minimum energy conformation from each run. They are useful for producing solutions very close to the global optimum in a reasonable amount of time. It is common practice to perform several runs in order to obtain a variety of different solutions and to investigate the nature of the energy surface.

### **Simulated Annealing**

Simulated annealing is a computational method that mimics annealing, the process in which the temperature of a molten substance is slowly reduced until the material crystallizes to give a large single crystal. The perfect crystal that is eventually obtained corresponds to the global minimum of the free energy. Simulated annealing is used to find the optimal or best solutions to problems which have a large number of possible solutions. Simulated annealing is a general purpose

optimization algorithm. It combines Markov-Chain Monte-Carlo methods (MCMC) ideas such as the Metropolis algorithm with a schedule for lowering temperature<sup>37</sup>.

In simulated annealing a cost function takes the role of the free energy in physical annealing and a control parameter corresponds to the temperature. To use simulated annealing in conformational analysis the cost function would be the internal energy. At a given temperature the system is allowed to reach 'thermal equilibrium' using a molecular dynamics or Monte Carlo simulation. At high temperatures the system is able to occupy high-energy regions of the conformational space and to pass over high energy barriers. As the temperature falls, the lower energy states become more probable in accordance with the Boltzmann distribution. At absolute zero, the system should occupy the lowest-energy state – the global minimum energy conformation. To guarantee that the globally optimal solution is reached would require an infinite number of temperature steps, at each of which the system would have to come to thermal equilibrium. Careful temperature control is required when the energy of the system is comparable with the height of the barriers that separate one region of conformational space from another. This is often difficult to achieve in practice and simulated annealing cannot guarantee to find the optimal solution. However, if the same answer is obtained from several different runs then there is a high probability that it corresponds to the true global minimum. Several simulated annealing runs may enable a series of low-energy conformations of a molecule to be obtained.

### **Clustering Algorithms and Pattern Recognition Techniques**

Molecular modeling programs generate large quantities of data that must be processed and analyzed. Many conformations search algorithms can generate conformations that are very similar, if not identical. Cluster analysis is used to select from the data a smaller, representative set of conformations for subsequent analysis. A common use of cluster analysis is in selecting a set of representative molecules from a large chemical database.

A cluster analysis requires a measure of the similarity between pairs of objects. A large number of cluster algorithms are available. Hierarchical clustering involves a series of iterations at each which the two closest clusters are identified and combined into a larger cluster. These methods produce a clustering that is independent of the order in which the objects are stored. Simple

implementations require an  $M \times M$  similarity matrix to be calculated, limiting their applicability when clustering large data sets. The Jarvis-Patrick method is a non-hierarchical clustering method that uses a nearest neighbors' approach. The algorithm can be used to cluster very large data sets. The K-means method is another non-hierarchical clustering method.

## 4.2 Tissue and Organ Modeling

Modeling of macromolecular interactions permits simulation of signal transduction into, across, and out of cells. With multi-cellular models, it is possible to investigate chemical and mechanical processes that occur in tissues, ranging from the relatively simple lipid bilayers that generally modulate intracellular chemistry to the much more complex assemblages that make up such tissues. Molecules, cells, tissues--the next level of biological organization is that of the organ itself. Complex organ models attempt to take into account explicit organ geometry coupled to hydrodynamics, continuum mechanics, reaction-diffusion, radiation and discrete particle transport. Organ modeling, while still in its infancy will be the springboard for detailed modeling of the body's organs as a complete system.

Organ modeling requires coupling of models and a system integration approach to coupling the models. First the physical organization, from the molecule to the organ must be modeled. Next the integration of functional models: chemical, mechanical, electrical, metabolic, and thermal. Then the models must be extended across broader scales of time and space. For example to model the heart, the anatomy and morphology of the heart need to be represented in the geometry and structure of the continuum mechanics model. The environmental influences need to be captured in the boundary conditions. The biological processes of mass transport, growth, metabolism, energetics, motion, flow, and equilibrium must be expressed through and must operate under the conservation laws for mass, energy, and momentum adopted in the model. Finally, structure-function relations need to be embodied in equations that take into account the material properties of the mechanical system.

Tissue and organ modeling generally begins with converting biology image data into computational meshes. This requires reconstruction and animation of volumetric deformable objects. In the scope of a medical application the goal is to simulate the motion and the form

alteration of the organ. One current research goal is to develop better shape representations for the deformable structures found in biomedical image databases. A longer-term goal of this research is to build specialized, deformable organ shape models that learn the priors for a particular organ type. Typically digital images are used to capture the organ detail. Then, mesh grid generation software is used to analyze the data and reconstruct it into a computer model.

### **Organ Model Applications**

Applications include simulation of dynamic and deformable behavior of cancerous tissues and organs to be integrated in radiation dose evaluation. A computational model of the cardiovascular system is aiding researchers in understanding the fundamental biochemical, biophysical, electrical and mechanical functions of the normal heart. The model is also advancing understanding of the molecular and genetic origins of heart disease, the electrical and mechanical properties of blood flow in large and small blood vessels; and the development of potential approaches for new cardiovascular drugs. A virtual lung model, developed at the Department of Energy's Pacific Northwest National Laboratory (PNNL), may help predict the impact of pollutants on respiratory systems and provide new insights into asthma, as well as other pulmonary diseases<sup>38</sup>. Using the virtual respiratory tract, PNNL scientists can analyze the influence of various factors, such as the amount of pollutants or length of exposure, on healthy versus diseased lungs by manipulating the computer model. With the model they can begin to simulate how gases, vapors and particulates may act differently within lungs of people suffering from cystic fibrosis, emphysema and asthma.

### **Heart Model Examples**

Heart models are the most advanced organ models. Two leading models are discussed here: the Peskin/McQueen model and the Cardiac Mechanics Research Group at University of California San Diego (UCSD) model. MetaCenter researchers Charles Peskin, David McQueen, and their group at the Courant Institute of Mathematical Sciences of New York University developed their heart model to help design improved artificial heart valves<sup>39</sup>. Their 3D model combines tissue and fluid mechanics models and permits the examination of the performance of modeled artificial heart valves for any of the four valves of the human heart. To solve both the fluid mechanics and elasticity problems simultaneously they use the immersed boundary method with

formal second-order accuracy<sup>40</sup>. This is essentially a second-order Runge-Kutta method. The heart is modeled as a set of elastic fibers immersed in an incompressible fluid, which avoids the complexities of applying boundary conditions on the moving location of the heart walls. For greater realism, they try to make the model fibers follow the same paths as muscle and collagen fibers in the real heart muscle and valves.

High resolution is also needed to make the Reynolds number (Re) realistic. To use a realistic Re for blood flow in the heart (about 500) requires a substantial refinement of the mesh, possibly by a factor of 25 in each spatial direction. The flow pattern of blood in the heart is not very sensitive to the Reynolds number, and improvements in numerical methodology, such as local mesh refinement near boundaries or the use of entirely grid-free methods, may make it possible to avoid the extreme computational requirements implied by such a refinement of a uniform grid. Nevertheless, a fully satisfactory computation of blood flow in the heart will require a substantial increase in computer power, Peskin says--possibly as great or greater than the increase that was needed to move from two to three dimensions.

Increased computer power is needed not only to do the current computation more correctly, but also to bring in additional phenomena that are highly relevant to blood flow in the heart. Two examples of such phenomena are the electrical activity that coordinates and controls the heartbeat and the dynamics of the blood clotting process, which is important in evaluating the function of prosthetic cardiac valves. Models of these phenomena are being developed separately from the model of cardiac mechanics described here. Microscopic and macroscopic models of the clotting process are being developed by Aaron Fogelson, a former student of Peskin's now at the University of Utah. Ultimately, Peskin and McQueen hope to combine such models with their mechanical model to increase its realism and predictive power. Again, a dramatic increase in computer power would be required.

The model's major algorithm and approaches are: the immersed boundary method; Navier-Stokes equations solved on cubic lattice through the use of FFT representations of the velocity and pressure; fiber equations solved on Lagrangian framework; and interface equations that

allow for a projection of the fiber forces onto the fluid lattice through a "smooth" delta function and interpolation of the fiber nodal velocities from fluid lattice.<sup>41</sup>

Modeling the heart presents many challenges. Beyond the usual difficulties of modeling fluid flows within rigid boundaries, the heart walls and heart walls and valves move and interact with the flow, both driving and responding to it. And not only is the heart muscle elastic, it is active--contracting and relaxing, with elastic properties that change during the contraction-relaxation cycle.

Katherine Yelick, a computer scientist at UC Berkeley, is leading a NPACI alpha project to make a generic immersed boundary code that will run on distributed parallel machines<sup>42</sup>. This NPACI alpha project is working toward an end-to-end demonstration of how a modern parallel language and compiler, Titanium, running on Blue Horizon with improved equation solvers and algorithms for handling adaptive computational grids, can support an important scientific application--simulating blood flow in the human heart. Yelick is porting the Titanium language, which provides greater support for parallel computing, to Blue Horizon, as well as porting the immersed boundary code to Titanium and developing scalable solver technology for uniform grids.<sup>43</sup> Colella is developing improved algorithms for handling adaptive computational grids, particularly for flows modeled by the immersed boundary method. Baden is developing communication support based on Kernel Lattice Parallelism (KeLP) for grid-based computation on Blue Horizon. Saltz is hardening the Titanium front-end for the Active Data Repository (ADR) storage facility to handle the immense data sets generated in the realistic simulations. Peskin identifies intelligent adaptive mesh algorithms that will "zero in" as the flow evolves during the simulation on the computationally challenging areas where the flow is more complex, such as near the delicate valve leaflets, as a critical requirement.

As with all the above methods that use computational meshes, the Cardiac Mechanics Research Group (CMRG) [Department of Bioengineering](http://cmrg.ucsd.edu/) and the [Whitaker Institute for Biomedical Engineering](http://cmrg.ucsd.edu/) at [UCSD](http://cmrg.ucsd.edu/) <http://cmrg.ucsd.edu/> heart model integrates structure and function as well as theory and experiment by means of the finite element method (FEM). The group uses a prolate spheroidal coordinate system to accurately represent both the compact shape and muscle fiber architecture of the heart's muscle walls. Various parameters, derived from laboratory

research, are introduced to the continuum model for comparative study. For instance, the CMRG members are working with both anatomic elements from rabbit hearts which have been histologically processed, and sections revealing the orientations of the muscle fibers. A second project relates to the differences in heart muscle structure between normal and brittle-boned mice suffering from osteogenesis imperfecta (OI) because of a deficiency in the protein collagen. The finite element models showed that OI mice develop variations in the residual stresses and muscle fiber structure which constitute beneficial adaptations to the deficiency of collagen.

As far as the human heart is concerned, the CMRG investigators study the relationships between the cellular and tissue structure of the ventricular myocardium as well as the mechanical and electro-physiological function of both the intact and affected organ. In ongoing projects, the mechanisms of ventricular mechano-electric feedback, the alterations during ventricular hypertrophy, and the flow-function relations during myocardial ischemia are unveiled. In collaboration with the Cleveland Clinic Foundation and the University of Auckland in New Zealand, the researchers in the Cardiac Mechanics Research Group explore the potential of a revolutionary surgical method for patients with severe heart failure. Through the combination of computational modeling with magnetic resonance imaging, the research is to predict which patients effectively can be rescued, using surgical ventricular reduction.

In an effort that includes applications of bioinformatics, the CMRG supports *Continuity 5.5* a computational tool for continuum problems in bioengineering and physiology, especially those related to cardiac mechanics and electrocardiology research. In addition to continuum modeling, *Continuity 5.5* has facilities for least-squares fitting of parametric models to experimental measurements from diverse sources including gross anatomy, histomorphology, 3-D medical imaging, and physiological and biomechanical testing. *Continuity 5.5* is component-based using a very high-level object-oriented scripting language for component integration. Executables for *Continuity 5.5* can be downloaded free from the group's website for academic research purposes. <http://cmrg.ucsd.edu/cgi-bin/cmrg/downloads/selection.cgi>.

### 4.3 Systems Biology

Systems biology aims at system-level understanding of biological systems. Molecular biology is mainly focused on identification of genes and functions of their products, which are components of the system. The next major challenge is to understand at the system level biological systems that are composed of components revealed by molecular biology<sup>44</sup>. The goal is to understand biological systems within a consistent framework of knowledge built up from the molecular level to the system level. Understanding biology at the system level – not only gene networks, but also protein networks, signaling networks, metabolic networks and specific systems such as the immune system or neuronal networks will be a major driver of HPCS requirements in the coming years. Understanding biological systems requires:

- Identification of the structures of the system – primarily regulatory relationships of genes and interactions of protein that provide signal transduction and metabolism pathways, as well as the physical structure of organisms, cell, organelle, chromatin and other components. Both the topological relationship of the network of components as well as parameters for each relation needs to be identified. Identification of gene regulatory networks for multicellular organisms is even more complex as it involves extensive cell-cell communication and physical configuration in 3-D space.
- Analysis of system behavior – once a system structure is identified, its behavior needs to be understood
- A method to control the state of biological systems
- Design of biological systems with the aim of providing cures for diseases

Simulations need to be able to simulate gene expression, metabolism and signal transduction for a single and multiple cells. The simulations must be able to simulate both high concentrations of proteins that can be described by differential equations and low concentrations of proteins that need to be handled by stochastic process simulation. Some efforts on simulating a stochastic process (McAdams and Arkin, 1998) and integrating it with high concentration level simulation are underway. In some cases the model requires not only gene regulatory networks and metabolic networks, but also high-level structures of chromosomes such as heterochromatin structures.

The simulations need to be coupled with parameter optimization tools, a hypothesis generator and a group of analysis tools. The algorithms need to be designed precisely for biological research. For example, the parameter optimizer needs to find as many local and global minima as possible because there are multiple possible solutions of which only one is actually used. The assumption that the most optimal solution is used in an actual system does not hold true in biological systems. The tools and analysis required are:

- A database for storing experimental data
- A cell and tissue simulator
- Parameter optimization software
- Bifurcation and systems analysis software
- Hypotheses generator and experiment planning advisor software, and
- Data visualization software

### **Systems Biology Computing Overview**

Cell biology is difficult to handle computationally. Cell signaling, cell motility, organelle transport, gene transcription, morphogenesis and cellular differentiation cannot easily be accommodated into existing computational frameworks. Conventional approaches using the numerical integration of continuous, deterministic rate equations can provide useful when systems are large or when molecular details are of little importance. However when the resolution of experimental techniques increases, conventional models become unwieldy. Difficulties include the importance of spatial location within the cell, the instability associated with reactions between small numbers of molecular species and the combinatorial explosion of large numbers of different species. For example, signaling pathways commonly operate close to points of instability and frequently employ feedback and oscillatory reaction networks that are sensitive to the operation of small numbers of molecules. Gene transcription is controlled by small assemblies of proteins operating in an all-or-none fashion, so that whether a specific protein is expressed or not is to some extent a matter of chance<sup>45</sup>. Stochastic methods are being used. The idea is to represent individual molecules rather than the concentrations of molecular species and to apply Monte Carlo methods to predict their interactions. In the stochastic

modeling approach, rate equations are replaced by individual reaction probabilities and the output has a physically realistic stochastic nature. Techniques are available by which large numbers of related species can be coded in an economical fashion and key concepts such as signaling complexes and heat-driven flipping of protein conformations can be embodied in the program<sup>46</sup>. (Shimizu and Bray, p. 215).

### **Systems Biology Analysis Methods**

Commonly used analysis methods for systems biology are bifurcation analysis, metabolic control analysis and sensitivity analysis. The [ERATO Systems Biology Workbench](#) project is to create an integrated software environment that permits sharing of models and resources between simulation and analysis tools for systems biology. The initial focus is on achieving interoperability between seven leading simulation tools: [BioSpice](#)<sup>47</sup>, [DBSolve](#)<sup>48</sup>, [E-Cell](#), [Gepasi](#)<sup>49</sup>, [Jarnac](#)<sup>50</sup>, [StochSim](#)<sup>51</sup>, and [Virtual Cell](#)<sup>52</sup>. As part of the effort, the project has also developed a model description language, the [Systems Biology Markup Language](#) (SBML) that can be used to represent models in a form independent of any specific simulation/analysis tool. SBML is a versatile and common standard that enables the exchange of data and modeling information among a wide variety of software systems<sup>53</sup>. It is an extension of XML, and is expected to become the industrial and academic standard of the data and model exchange format.

At a very abstract level, a cell can be divided into two general subnetworks, a regulatory network and a metabolic network<sup>54</sup>. These networks possess very different characteristics. The metabolic network is mainly occupied with substance transformation to provide metabolites and cellular structures. The regulatory network's main task is information processing for the adjustment of enzyme concentrations to the requirements of variable internal and external conditions. This network involves the use of genetic information.

The Virtual Laboratory uses a process modeling tool [PROMOT](#), originally designed for application in chemical engineering. It allows for the computer-aided development and implementation of mathematical models for living systems<sup>55</sup>. For the numerical analysis of the resulting models the simulation environment [DIVA](#)<sup>56</sup> is used. DIVA deals not only with large-scale differential-algebraic systems which arise in chemical process engineering but also in the mathematical modeling of complex cellular networks. Inside DIVA many different numerical computations can be performed based on the same model, including dynamic and steady state

simulation, parameter estimation, optimization and the analysis of nonlinear dynamics. There are currently four methods of special interest for cellular models:

- Dynamic simulation of the models with different integration algorithms.
- Sensitivity analysis for parameters with respect to experimental data.
- Parameter identification according to experimental data.
- Model-based experimental design.

Most numerical algorithms in DIVA are taken from professional numerical libraries like HARWELL and NAG. The system also has additional methods like steady state continuation and bifurcation analysis. The visualization and postprocessing are done using MATLAB. Stochastic modeling is an approach to modeling phenomena such as intracellular signaling and gene expression. The conventional approach of representing biochemical reactions by continuous, deterministic rate equations cannot be easily applied to intracellular processes based on multiprotein complexes or those that depend on the individual behavior of small numbers of molecules<sup>57</sup>. Two stochastic approaches are STOCHSIM and the Gillespie approach<sup>58</sup>.

Software packages that allow kinetic performance of enzyme pathways to be represented and evaluated quantitatively: (flux-analysis programs, often aided by metabolic control analysis) are [GEPASI](#), MIST<sup>59</sup>, and [SCAMP](#)<sup>60</sup>.

Software suites for the recording and analysis of electrical data, the simulated performance of individual axons and the investigation of networks or nerve cells (neurobiology and cortical activity), are: [GENSIS](#)<sup>61</sup> and [NEURON](#)<sup>62</sup>.

#### **4.4 Computational Biology Case Study: The Cardiac Mechanics Group at UCSD**

We spoke with Dr. Taras Usyk and Sarah Healay. Dr. Taras Usyk is an Assistant Project Scientist and Sarah Healy is an advanced graduate student. Both are researchers in the Cardiac Mechanics Group directed by Andrew McCulloch. [Andrew McCulloch](#) is Professor of Bioengineering at the [University of California San Diego](#). He is also a member of the [Whitaker Institute of Biomedical Engineering](#), the UCSD/Salk Institute for Molecular Medicine and the

Center for Research in Biological Structure, a Senior Fellow of the [San Diego Supercomputer Center](#), and Director of the [BioNOME Resource](#) at the San Diego Supercomputer Center. Dr. Usyk and Sarah Healy collaborate on structurally and functionally integrated numerical models of cardiac electromechanics using the finite element method. Their goal is to perform whole organ simulations with biophysically detailed systems models involving over a hundred thousand degrees of freedom.

The members of the Cardiac Group shared some the computational demand associated with modeling the heart mechanics and electro-physiology using finite element methods. Electrophysiological models of the heart require small temporal and spatial scales, determined by the system of ODEs that comprise the cellular model. An operator splitting algorithm allows ODE and PDE systems to be solved separately with updates occurring every half timestep. Of these, the ODEs dominate with 90-95% of compute time being spend on them. However, this system is also data parallel and is where they are focusing their efforts. An example of an electrophysiological model that incorporates a mere  $2 \times 1 \times 0.5 \text{ cm}^3$  volume of the heart ventricle uses 1024 finite elements in a tricubic spline approximation; state variables are evaluated at over 10,000 points in the volume. This represents  $1/24^{\text{th}}$  of the whole rabbit ventricle.

The finite element code currently requires 300 Mbytes of memory to run; about 250 Mbytes are used by the linear solver and the rest for the input file. Memory requirements scale roughly linearly with the mesh size. The code is written in a combination of python and Fortran, with the use of a linear solver written in C. It currently takes 10 hours to solve the wedge mesh to 1 second, with a time step of 0.1 ms on a single processor Pentium 4 running Linux. They would expect close to linear speedup on a parallel machine.

The mesh size for mechanical models of the heart is determined by the ability of solvers to converge to stable solutions. While mechanical models typically require a fraction of the number of mesh points that electrophysiological models require, they have larger memory requirements, requiring 4-5 Gbytes. This presents a problem when running on machines that have 2-4 Gbytes of memory; it often means simplifying components of the model.

The Cardiac Group runs simulations a heterogeneous coding and system environment. For example, they currently embed non-native code into a code base, for example embedding Fortran into Python. This introduces technical issues with latency and compile efficiency. There is latency associated with handoff of I/O operations. Native and non-native code has to be compiled and debugged separately.

The members of the Cardiac Group come from various backgrounds, mainly in the biological sciences. They identified tools that would assist them improve their productivity in an intensive computational environment with limited formal coursework in the computer sciences. Productivity could be drastically improved with is better debuggers and standard options on compilers. Relying on “print” statements for debugging is time consuming. O/S stability is also an issue. Some versions of linux are more likely to crash than others. Productivity could be increased if there where better strategies for dealing with multiple users vying for the same memory. Currently, users are simply kicked off when too many users are on client/server systems. This results in loss of data and time.

There was a general consensus that demand for more computing power will never saturate. Currently, heart simulations run for only several to a couple of heart beats. Increased computational power will allow researchers to run to simulate minutes instead of seconds, using more accurate models and including pathologies and cellular level information to gain more robust information about heart function.

## **4.5 Computational Biology Needs**

### **Interviewees**

#### **Taras Usyk, Ph.D. and Sarah Healy**

Dr. Taras Usyk is an Assistant Project Scientist and Sarah Healy is a 3<sup>rd</sup> year graduate student. Both are researchers in the Cardiac Mechanics Group directed by Andrew McCulloch. Dr. Usyk and Sarah Healy collaborate on structurally and functionally integrated numerical models of cardiac electromechanics using the finite element method.

**Adam Arkin, Ph.D.**

Adam Arkin is an Assistant Professor of Bioengineering at the University of California, Berkeley. He is also a Faculty Scientist in Physical Biosciences at the Lawrence Berkeley National Laboratory. He is one of the central developers of BioSPICE and the director of the Virtual Institute of Microbial Stress and Survival (<http://vimss.org>).

**James B. Bassingthwaite, Ph.D.**

James Bassingthwaite is a Professor in the Department of Bioengineering at the University of Washington. He is the director of the National Simulation Resource Facility for Circulatory Transport and Exchange. NSR was created with a focus on studying complex biological systems and networks involved in the transport and exchange of solutes and water in the microvasculature, within whole organs, and within the whole body.

**Steinar Hauan, Ph.D.**

Steinar Hauan is a Professor of Chemical Engineering in the Biomedical Engineering Department at the Carnegie Mellon University. Professor Hauan's research is in the area of computer-aided process design and analysis of complex chemical systems.

**Juan Cebal, Ph.D.**

Juan Cebal discussed the research he and the research team including Rainald Loehner, and Orlando Soto, George Mason Univ.; and Peter L. Choyke and Peter J. Yim, National Institutes of Health. The application he discussed is an image-based finite element model of hemodynamics in stenose carotid, a methodology to construct patient-specific, anatomically and physiologically realistic finite element models of blood flows.

**Requirements**

1. Computational approaches that apply asynchronous agents that collaborate to arrive at a solution to complex problems are often multi-threaded and require the development of distributed algorithms without central control of agents and complex adaptive systems to monitor CPU time.

2. There are holes in existing algorithms that handle remote processes; current RPC code is not robust enough. Current versions of MPI are not fault tolerant and do not scale well to grid processing. MPI was not designed to be fault-tolerant for asynchronous system; rather, it was implemented to enforce synchronization and would thus never get (or need) the type of redundancy and fault tolerance necessary for large scale, asynchronous processing. Management tools are needed in a distributed environment to recover, or at least ignore, failures in communication.
3. Compiler speed and CPU types compare differently for different systems. There is no correlation between compilers and applications. The work done per cycle on the same hardware varies with compilers. Speed and performance varies across systems and compilers. It would help if someone had a benchmark library for different types of calculations. This would enable users to better evaluate what machines they should purchase and use, based on their specific applications.
4. A considerable amount of time is spent compiling code and it is often easier to reproduce code rather than re-use code written by another researcher using a different compiler. Compile time could be reduced drastically by designing compilers that are efficient across platforms but compile code from anywhere without library dependencies or with very well defined and packaged library dependencies. Code diagnostic tools need to be updated. Code has become sufficiently complicated that a real tool to for designers and project managers to visualize large coding projects would improve productivity.
5. Two important classes of algorithms that will play a large role in biomedical research are finite element code and density functional calculations. It will be important to lower the barrier for researchers to use these tools and provide parallelized version of existing code.
6. PDE optimization will require algorithms with improved performance. There is a need, however, to integrate PDE based optimization with logic based optimization and to move this integration to large, complex systems.
7. Cellular processes take place on many time scales; different reactions have different characteristic rates. Including cellular mechanics introduces another set of times scales. Algorithm development and formal abstraction will probably be the most important

aspect of dealing with simulations involving multiple time scales. New algorithms should both be able to separate slow and fast time scales, with well-understood and defined approximation errors, and be numerically stable. Multi-scale models are extremely complex. They incorporate information from the molecular and cellular levels up to organ and systems levels. Realistic models of the circulatory and respiratory systems under stress, for example exercise, require a description of the cellular events that create demands for oxygen. Having cellular level equations together with circulatory exchanges makes the system very stiff. Brute force methods are possible, but computationally demanding. It is equally challenging, however, to simplify models by using the results from the cellular level as descriptors to drive the higher level equations; changes at the higher level, for example start and stop of exercise, must be communicated back to the basic model. The development of strategies to automate the switching from the simplified submodels to the more detailed realistic submodels is critical to the designing of efficient yet realistic models that encompass several hierarchical levels.

8. While there have been several attempts, there is still no good visualization tools for large scale, high dimensional data sets. An interesting challenge for researchers is to develop tools to represent complex chemical networks. Such tools might allow researchers to visualize network behavior and to map networks and their products, providing information about the state of the system as parameters are changed.
9. Parallelization of molecular dynamic code and simulations of reaction/diffusion processes is challenging because although there is a maximum diffusion within a time step (and there are hundreds of thousands of time steps), it is not possible to know how many molecules will enter from nearby processors. Processor boundaries introduce uncertainties in handling communications and the need to detect termination. Key requirements: random number generators; vastly improved reliability of systems; tools to support load balancing.
10. Embedding non-native code into a code base, for example embedding Fortran into Python, introduces technical issues with latency and compile efficiency. There is latency associated with handoff of I/O operations. Native and non-native code has to be compiled and debugged separately.

11. O/S stability is an issue. Some versions of linux are more likely to crash than others. Productivity could be increased with better strategies for dealing with multiple users vying for the same memory. Currently, users are simply kicked off when too many users are on client/server systems. This results in loss of data and time.

Computational biology—the unique mix of molecular biology and computer science—has come of age in recent years, earning status as a scientific discipline in its own right. As part of an effort to accelerate medical discovery to improve health, NIH has funded several centers focused on aggressively pushing the boundaries of current computational needs. These centers will focus on Physics-based simulations of biological processes, analysis and visualization of medical image data, scalable computational and organizational framework for conducting clinical research, and computational and mathematical approaches to the study of genes, cells, systems and whole brain. The domain of computational biology can be considered a set of overlapping atlases - sets of maps on different spheres of biological information that span many scales and modalities from genotype to phenotype. The concept of computational atlases can be understood as a database-like infrastructure that rests on mathematical advances in modeling and optimization. As the infrastructure for such a computational atlas is developed, it will be possible to begin to address large-scale modeling problems that before now have been intractable.

## Chapter 5: Drug Discovery

Discovering and developing any new medicine is a long and expensive process. A new compound must not only produce the desired result with minimal side-effects but must also be demonstrably better than existing therapies. Typically, the two key steps in drug discovery programs are the identification of “hit” molecules and lead series, or “leads”. A *hit* is a molecule that has some reproducible activity in a biological assay. A *lead series* comprises a set of related molecules that usually share some common structural feature and which show some variation in the activity as the structure is modified. This provides confidence that further synthetic modification to the lead series has a good chance of resulting in a drug candidate with the desired potency and selectivity, lack of toxicity and appropriate characteristics to enable it to reach its

target in vivo. Such a drug candidate will then enter the early stages of development, where further large-scale investigations are undertaken.

## 5.1 Drug Discovery Overview

Although high-throughput screening makes it possible in principle to test every available compound against every biological assay, there are number of practical reasons why this is not feasible:

- The large number of samples now available in many companies means that the overall expense can be significant
- Some assays cannot be converted to a high-throughput format and so have to be conducted using more traditional technique
- A significant proportion of the available samples might not be considered appropriate structures.

As a result, it is often necessary to identify subsets of compounds. Computational techniques play a significant role in determining which such subsets can be constructed, with various techniques being available depending upon the type of molecule to be screened, the information available to assist with the selection and the properties to be taken into account.

A wide variety of methods are used either individually or in combination to select compounds. 2-D methods use only information about the chemical structure of the molecule. 3-D methods use information about the molecules confirmation and properties dependent upon the confirmation. Some methods take into account information about the target protein or about other molecules that are known to be active at the target, whereas other methods are designed to produce diverse collections of compounds for more general screening.

Having tested a number of compounds, a model is usually constructed that relates the observed activity to the molecular structure. The model can then be used in the next iteration of the process. Many different kinds of models are used. A popular approach is to use statistical techniques to derive the model.

## **Substructure Searching**

Substructure searching is the most basic approach to identifying compounds of interest. Many organizations maintain databases of chemical compounds; some are non-proprietary and some are proprietary. A database may consist of large numbers of compounds, several hundred thousand is common. The American Chemical Society database contains more than 18 million compounds.<sup>63</sup> Most systems represent molecules as molecular graphs. A graph contains nodes, which are connected by edges. A subgraph is a subset of the nodes and edges of a graph. A key requirement for any chemical database system is that it can determine whether or not a new molecule is already present in the systems. A substructure search retrieves all the molecules from the database that contain the substructure. Substructure searching is known as subgraph isomerism – determining whether one graph is entirely contained within another. Even with the most efficient algorithms this is a relatively time-consuming process and so chemical database systems commonly use some form of screening method to rapidly eliminate molecules that cannot match the query. Such screens are frequently implemented using binary representations and so operate rapidly, especially if held in memory.

## **Binary Screening**

Two types of binary screening are used. In a structural key, each position in the bitstring corresponds to a particular substructure. If that substructure is present in the molecule, then the relevant bit in the molecule's key is set to 1. A predefined fragment dictionary is used to specify the substructures. As each molecule is added to the database a substructure search is performed for each fragment and the relevant bit assigned. Many different types of substructure can be incorporated, such as the presence or absence of particular elements, rings and common functional groups. It is also possible to assign bits which encode how many occurrences of a particular feature exist. Structural keys used by the MACCS and Isis systems from Molecular Design are the best known of this type of bitstring.

Hashing fingerprint is a second commonly used type of binary screening, and does not require a predefined fragment dictionary; it uses an algorithmic approach to derive the bitstring. The Hashing fingerprint method produces all possible linear paths of connected atoms through the molecule containing between 1 and a pre-defined number of atoms. Each path defines a pattern

of atoms and bonds which serves as the input to a pseudo-random number generator, which produces a set of bits which are then set to the value 1. The hashing process typically sets 4 or 5 bits per pattern. A bitstring might contain 1024 bits and after all paths have been examined a typical organic, drug-like molecule might have a total of 200-300 bits set to 1. Hashed fingerprints are used in a number of database systems and are particularly associated with the systems from Daylight Chemical Information Systems.

When using a bitstring screen, one first calculates the corresponding bitstring for the substructure query. Next, the query bitstring is compared with the bitstrings for all the molecules in the database. A molecule can only possibly match the query if it contains a 1 for every position in the bitstring where the query also has a 1. Well-designed screens can eliminate up to 99% of the molecular during this phase. After eliminating molecules that could not match the query, an atom-by-atom search for the molecules in conducted. One commonly used method is the Ullmann algorithm which represents the molecular graphs of both the query substructure and the potential molecular match by an adjacency matrix, which is a square, symmetric matrix such that the element ( $ij$ ) has the value 1 if atoms  $i$  and  $j$  are bonded, and zero otherwise. The Ullmann algorithm tries to find matrices  $A$  such that  $A(AM)^T$  is identical to  $S$ , where  $M$  is the adjacency matrix of the molecule and  $S$  is the adjacency matrix of the substructure<sup>64</sup>.

### **Database Searching – Conformational Properties and Functionality Features**

A 3D database search allows one to identify molecules that satisfy the chemical and geometric requirements of the receptor. A 3D database contains information about the conformational properties and functionality features of the molecules contained within it. There are two general types of 3D database searches. The choice of which to use depends on the information available about the target receptor. Pharmacophore mapping is used when an experimental structure of the target macromolecule is not available. Once a pharmacophore has been developed, it can then be used to find or suggest other active molecules. A pharmacophore refers to a set of features that is common to a series of active molecules. Such features are referred to as pharmacophoric groups, functional groups or molecules with similar physical and chemical properties such that they produce generally similar biological properties<sup>65, 66</sup> [Thornber 1979; Patani and LaVoie 1996]. A 3D pharmacophore specifies the spatial relationship between the groups. These relationships are

often expressed as distances or distance ranges but may also include other geometric measures such as angles and planes.

There are two problems to consider when calculating 3D pharmacophores. First, unless the molecules are all completely rigid, one must take account of their conformational properties. The second problem is to determine which combinations of pharmacophoric groups are common to the molecules and can be positioned in a similar orientation in space. More than one pharmacophore may be possible. Some algorithms can generate hundreds of possible pharmacophores, which must then be evaluated to determine which best fits the data.

### **Constrained Systematic Search**

Constrained systematic search address the problem of determining conformations in which the inhibitors can position multiple pharmacophoric groups in the same relative position in space. The constrained systematic search method of Dammkoehler, Motic and Marshall<sup>67</sup> showed that it is possible to determine what torsion angles of the rotatable bonds will enable conformations consistent with the previous results to be obtained.

### **Ensemble Distance Geometry**

Ensemble distance geometry can be used to simultaneously derive a set of conformations with a previously defined set of pharmacophoric groups overlaid. Ensemble distance geometry uses the same steps as standard distance geometry with the special feature that the conformation spaces of all the molecules are considered simultaneously<sup>68</sup>.

### **Clique Detection Methods**

It may be difficult to identify all possible combinations of the functional groups when many pharmacophoric groups are present in the molecule. Clique detection algorithms can be applied to a set of precalculated conformations of the molecules. Cliques are based upon the graph-theoretical approach to molecular structure. A clique is defined as a maximal completely connected subgraph. Find the cliques in a graph in NP-complete. Many algorithms have been devised for finding cliques, including the method of Bron and Kerbosch<sup>69</sup>. The algorithm can be described as:

- Generate a family of low-energy conformations for the molecules
- Use the molecule with the smallest number of conformations as the starting point
- Use each of its conformations in turn and the reference structure
- Compare each conformation of every other molecule with the reference conformations and the cliques identified
- Obtain the cliques for each molecule by combining the results for each of its conformations
- Combine those cliques that are common to at least one conformation from each molecule to give a possible 3D pharmacophore for the entire set

### **Maximum Likelihood Method**

The maximum likelihood method eliminates the need for a reference conformation, effectively enabling every confirmation of every molecule to act as the reference. The algorithm scales linearly with the number of conformations per molecule, thus enable a large number of conformations to be handled.<sup>70</sup> The algorithm can be described as follows:

- Generate a set of conformations for each molecule
- Consider all possible combinations of pharmacophore features exhaustively
- Identify possible geometric arrangements of the features ins 3D space
- Score and rank according to how well the configuration describes the set of active molecules

## **5.2 Molecular Docking**

Molecular docking attempts to predict the structure of the intermolecular complex formed between two or more molecules. Most docking algorithms are able to generate a large number of possible structures, and so they also require a means to score each structure to identify those of most interest. The docking problem involves many degrees of freedom. There are six degrees of translational and rotational freedom of one molecule relative to the other as well as the conformational degrees of freedom of each molecule.

Various algorithms have been developed to tackle the docking problem and can be characterized by the number of degrees of freedom they ignore. The simplest algorithms treat the two molecules as rigid bodies and explore only the six degrees of translational and rotational freedom.

To perform conformationally flexible docking the conformational degrees of freedom need to be taken into account. All of the common methods for searching conformational space have been incorporated at some stage into a docking algorithm. Monte Carlo methods have been used to perform molecular docking, often in conjunction with simulated annealing.<sup>71</sup> Genetic algorithms can also be used to perform docking<sup>72</sup>, as well as distance geometry. An approach that is used by a number of programs involves the incremental construction of the ligand<sup>73</sup>. A typical incremental construction algorithm first identifies one or more base fragments within the ligand. The base fragments are docked into the binding site and may then be clustered to remove similar orientations. Each docked orientation of the base fragment(s) then represents the starting point for the conformational analysis of the rest of the ligand.

The ideal docking methods would allow both ligand and receptor to explore their conformational degrees of freedom. Molecular dynamics simulation of the ligand-receptor complex is one way to do this. However such calculations are computationally very demanding and typically used for refining structures produced using other docking methods.

Most docking algorithms generate a large number of potential solutions. Some of these can be rejected immediately because they have a high-energy clash with the protein. The rest are assessed using some scoring function. Many of the scoring functions attempt to approximate the binding free energy for the ligand binding to the receptor. Molecular mechanics is also widely used to calculate the energy of interaction. The calculation can be speeded up by pre-calculating electrostatic and van der Waals potentials on a regular grid that covers the binding site. The computational effort required to calculate the energy of interaction between ligand and protein is then linear in the number of atoms in the ligand, rather than being proportional to the product of the number of ligand atoms multiplied by the number of protein atoms.<sup>74</sup> Combining the results

from more than one scoring function has been shown to give better results than using individual scoring functions on their own, an approach referred to as consensus scoring.<sup>75</sup>

### **Software for automated docking**

**GOLD:** <http://www.ccdc.cam.ac.uk/prods/gold.html>

**AutoDock:** <http://www.scripps.edu/pub/olson-web/doc/autodock/>

**DOCK:** <http://www.cmpchem.ucsf.edu/kuntz/dock.html>

**DockVision:** <http://www.dockvision.com>

**FlexX:** <http://cartan.gmd.de/FlexX>

**ICM:** <http://www.molsoft.com/products/modules/dock.htm>

### **Protein-Ligand Docking in Drug Design**

The first step of the drug design is to identify the lead structure, a small molecule which binds to a given target protein. The docking problems can be categorized as:

Given two molecules with detailed 3-D structures:

- Binding properties: bond strength and binding-complex morphology.
- Protein-Protein or Protein-DNA docking: rigid-body docking, i.e., fixed overall shapes.
- Protein-Ligand docking: the ligand is not fixed in its overall shape

Since most drugs are small molecules, protein-ligand docking is of great interest in pharmaceutical. The basic docking idea is to represent the active site by a set of spheres and then perform sphere matching<sup>76</sup>. There are two main algorithms, which are described below:

#### ***Algorithm 1: SPHGEN***

- calculate the molecular surface
- generate spheres covering the active site
- cluster spheres, remove very similar ones
- radius too large
- select clusters defining the active site

- color spheres by properties

### **Algorithm 2: MATCH**

(calculate a matching between ligand atoms L and protein spheres K)

- two matches  $(l_1, k_1), (l_2, k_2)$  are *distance-compatible* if  $|d(l_1, l_2) - d(k_1, k_2)| \leq \varepsilon$
- search for matchings  $M_{i,j} = \{(l_i, k_j)\}$  with  $\max |d(l_1, l_2) - d(k_1, k_2)| \leq \varepsilon$
- Matching-Graph: nodes  $L \times K$ , edges between distance compatible nodes
- Matchings are cliques in the matching graph (cliques = completely connected subgraphs)

### **Rigid-Body Protein-Ligand Docking**

With rigid-body protein-ligand docking, the protein and the ligand are assumed to be rigid. The first and most widely used rigid-body protein-ligand docking algorithm is DOCK<sup>77</sup>. The algorithm can be described as follows:

- a set of spheres is created inside the active site,
- the sphere represents the volume which could be occupied by the ligand molecule
- the algorithm searches for ligands (represented by spheres) that match the spheres describing the active site

Docking methods use receptor-ligand interactions to suggest binding modes. This is accomplished by identifying regions of binding site liable to interact in a given way *e.g.* hydrophobic regions or H-bonds. These interactions are clearly important, but other factors also affect binding. *Scoring functions* attempt to use all such factors to rank docked complexes in order of tightness of binding. Different scoring functions vary in which terms they treat and exact form of treatment.

DOCK comes with a very simple scoring function to complement simple shape-based docking algorithm. DOCK ignores solvation, conformation and entropic effects completely. It uses molecular mechanics method to estimate binding free energy – equivalent to binding enthalpy in this case. DOCK uses AMBER force field for binding electrostatics & sterics *i.e.*

$$\Delta G_{\text{bind}} \approx \Delta E_{\text{elec}} + \Delta E_{\text{vdw/ster}}$$

$\Delta_{\text{Evdw/ster}}$  includes attractive van der Waals interactions and repulsive steric clashes. These are calculated from the standard Lennard-Jones 6-12 potential using pairwise atom-atom terms.

The electrostatic term in DOCK is taken as a simple sum of charge-charge interactions. Charges are estimated by Gasteiger's electronegativity equalisation scheme – fast route to charges from 2D structure only. Dielectric shielding can be applied to above equation to model the shielding of charges by each other. The value of  $\epsilon$  varies for different receptors.

The simplicity of DOCK is attractive – it is very quick to evaluate & easy to interpret. Calculation can be speeded up by evaluating electrostatic & conformation 'fields' on a grid within the binding site. The score for a given ligand is then easy to calculate from atomic positions in the grid. The lack of entropic and conformation effects means DOCK is only applicable to series of similar ligands. In spite of this limitation, DOCK is remarkably successful.

A different approach to empirical scoring is the 'Potential of Mean Force' (PMF) function of Muegge & Martin. Atom types for important interactions in complexes are identified. The strength of atom-atom interactions is found by regression against binding energies including distance-dependence terms. This includes solvation, entropic *etc.* effects implicitly, and therefore is very fast.

Recent studies suggest that no single scoring function works for every problem. Two main measures of quality are used:

- comparison with ranking from experimental binding energies
- agreement with X-ray structure (RMSD)

Certain interactions are better represented by different scoring functions. Simple DOCK approaches can work better than more complex ones. This leads to *consensus scoring*, where several scoring functions are used at once. The simplest approach is to take the average predicted binding energies – but this doesn't work well as results are not always on same scale.

Rank order (*i.e.* 1st,2nd,3rd,...) has been shown to be better than predicted DGbind. Several statistics for consensus scoring have been proposed from rank. The first study used DOCK, GOLD, FlexX and PMF to rank 15 ligands from 1st-15<sup>th</sup>. Best criteria found to be ‘worst-best rank’ and ‘rank-sum’. Worst-best drops the worst of the 4 ranks and takes next worst, while rank-sum adds the remaining 3 ranks *e.g.* ligand has ranks 3, 5, 6 & 12 – worst best is 6th and rank-sum is 14. This method is known as CScore – and is implemented in several packages now.

### **Flexible ligand docking**

GOLD, a genetic optimization for ligand docking is a program developed by Gareth Jones at the University of Sheffield (Sheffield, UK) in collaboration with Glaxo Wellcome (London) and the Cambridge Crystallographic Data Centre (CCDC; Cambridge, UK) where the technique is applied to the problem of docking ligands to protein binding sites<sup>78</sup>. One *chromosome* describes the conformation of the ligand and selected protein side chains by defining the torsion angle of each rotatable bond. Another *chromosome* stores a mapping between hydrogen bond partners in the protein and the ligand. 3D structures are generated from these two chromosomes. A scoring function that evaluates the hydrogen bond, ligand internal energy and van der Waals energy is applied as the fitness function. The GOLD docking method also has its own scoring function, which is slightly more sophisticated than DOCK. Rather than a simple electrostatic term, GOLD models the H-bonds in a complex. Careful studies of how small molecules interact and crystallise give geometry/energy rules for different H-bonds. GOLD is slower to evaluate than DOCK, but its better search capability results in comparable performance.

FlexX is perhaps the most complex scoring function currently available. All the scoring functions discussed so far attempt to calculate binding energy directly. All employ some version of the ‘master equation’ which partitions DGbind. An alternative approach is to find an ‘empirical scoring function’ from a database of binding energies. Empirical scoring functions are a form of QSAR, with DGbind in place of activity. These calculate properties of ligands and train the QSAR/scoring function from known values using linear regression and/or PLS. Properties used include polar/non-polar Solvent Accessible Surface Area (SASA), H-bond donors and acceptors and number of rotatable bonds.

## Docking by Simulation

Molecular dynamics simulations use the force field to calculate the forces on each atom of the whole system. Following classical mechanics, velocities and accelerations are calculated and the atoms are moved slightly with respect to a given time step. Simulated annealing is another optimization algorithm that avoids getting into local minima but lacks physical interpretation of the simulation itself.

## 5.3 *De novo* Drug Design

Docking/scoring finds active molecules from a list of possible ligands. *De novo* drug design attempts to find new structures rather than comparing new ones. While database searching is an attractive way to discover new lead compounds, database searching does not provide molecules that are structurally novel. In addition, many databases are biased towards particular classes of compounds, and so limit the range of structures that can be found. In *de novo* design, the 3-D structure of the receptor or the 3D pharmacophore is used to design new molecules. The starting point is a receptor site from X-ray or modelling. However, instead of possible ligand molecules, a database of common & realistic fragments is searched for a fit with binding site. Fragments are chosen to give good shape and interaction overlap with binding site. Scoring functions can be used to define which fragment(s) are best suited. Fragments are then re-combined somehow to give possible drug leads. There are two basic types of *de novo* design algorithm: ‘outside-in’ or ‘inside-out’. Both look to grow a fragment within the binding site, but differ in how fragments are chosen and molecule re-combined. These will ideally converge to the same (or similar) solution given the same database and scoring function. This is a relatively new approach, so not yet clear if either has any advantages.

The outside-in method<sup>79</sup> finds fragments which bind tightly to regions of active site, which are then combined to real molecules. Initially the binding site is sampled for ‘*site points*’ where interactions could occur. Next, fragments are placed on site points and scored with some function. The scoring function and systematic search can usually identify binding fragments. This is only half the problem – combining fragments is not trivial. One solution is to use a database of common connectors, and match geometry of fragments to known linker groups.

Alternatively a ‘skeleton’ can be grown between fragments using rings and/or acyclic C—C bonds.

The inside-out method takes the opposite approach. First one starts with a central ‘scaffold’ fragment and incrementally grows fragments off this. The scaffold fragment should be rigid and tightly bound, but this is not always obvious. Each time a fragment is added, the resulting molecule is re-docked into binding site. Once docked the new molecule’s binding energy can be predicted using one or more scoring functions. The search stops either when no further improvement found, or when binding is tighter than some pre-defined cutoff. Repeated docking and scoring can be slow, but can discover new ligands.

The main goal of rational or *de novo* design is to find an active compounds for further development, or *leads*. Lead optimisation is equally important in overall drug development process. Enhancement of activity is a major goal, but better solubility, delivery/distribution, toxicity & synthesis also important. This often takes the form of ‘virtual screening’ of possible targets before any synthesis.

## **Chapter 6: Computer-aided diagnostic imaging and image-guided interventions**

Recent advances in imaging research have shown the potential to change many aspects of clinical medicine within the next decade. Image-guided therapy is growing rapidly. Increasingly diseases are being diagnosed and treated using less invasive, more sophisticated imaging and image-guided procedures. Major new areas of research focus on development of the molecular, functional, cellular, and genetic imaging tools, aided by new information technology and image fusion/integration capabilities.

The multidisciplinary field of image-guided therapy and surgery has become increasingly refined with application of techniques such as MRI, CT, and ultrasonography. Image-guided surgery

brings powerful technologies into the operating room by applying advances in computer science and engineering. The simultaneous combination of direct vision and imaging is possible with intraoperative MRI.<sup>27-32</sup> Open-configuration MRI systems guide, plan, and direct multiple procedures from biopsies to percutaneous interventions and neurosurgery. Use of MRI to guide biopsies of lesions that cannot otherwise be detected and to direct therapy is a powerful application of this technology.

Functional imaging (functional MRI/SPECT/PET) makes it possible to map brain function directly in the operating room.<sup>28</sup> Functional MRI allows identification of the brain area by function, such as the speech center or motor cortex, and the surgeon can avoid damage to such critical areas. For certain interventions (e.g., biopsies, tumor resection, directed therapies), this imaging information enhances the ability to apply sophisticated imaging techniques to surgery. Intraoperative MR guidance for neurosurgery improves precision of tumor resection, particularly when high-resolution MRI images are combined with functional MRI, SPECT, and MR angiographic data. Further advances in MR-guided interventions, biopsies, ablations, and surgery are needed to expand their capabilities.

Computer-aided diagnostic imaging and image-guided interventions are used for monitoring of disease progression, diagnosis, preoperative planning and intraoperative guidance and monitoring. Postprocessing adds value to medical images. However, successful postprocessing requires complex and optimized processing systems.

The increasing complexity of information available from image data sets increases demand on the diagnostic skills of radiologists. Two ways to improve diagnostic performance are by improving the radiologist's accuracy and by increasing the utility of diagnostic decisions. The ability to perform multimodal image fusion (e.g., combine data sets from PET and CT or SPECT and MRI) increases complexity and also requires innovative methods for increasing diagnostic accuracy, such as feature analysis and computer-aided diagnostic tools. Statistical prediction rules are a form of computer-based decision support that improves diagnostic accuracy. Such rules can enable analysis of more than 20 variables on a mammogram and combine the results to provide an estimate of the probability of cancer. These tools are powerful and can improve the

quality and accuracy of diagnostic techniques, as illustrated by application of MRI for staging prostate cancer.

## 6.1 Image Processing: Segmentation

Key issues for digital imaging are segmentation and registration. Signal processing techniques are used to enhance features and generate the desired segmentation. Results of the segmentation are aligned to other data acquisitions and to the actual patient during surgical procedures. Results of the segmentation are visualized using different rendering methods.

### Feature Enhancement

Image data is filtered prior to segmentation to reduce the noise level and to emphasize image structures of interest. Segmentation of MR images often uses anisotropic diffusion for enhancing the gray-level image structure prior to segmentation. By smoothing along structures and not across, the noise level can be reduced without severely blurring the image. Steerable filters that conform to the local structure adaptively are often used.

Convolution involves multiplication and summation of filter kernel coefficients with signal voxels, over the local area that the filter supports. Since the result in each voxel can be calculated independently, these calculations can be done in parallel and thus the speedup for convolution is linear with the number of CPUs. For large filter kernels (e.g., 9x9x9 voxels), it is more efficient to calculate the result of a convolution using the Discrete Fourier Transform (DFT).

For example, FFTW is a software package developed at MIT. FFTW is a C subroutine library for performing the Discrete Fourier Transform (DFT) in one or more dimensions. An MPI version of the FFTW routines is available which makes it possible to perform the FFT calculation on distributed memory machines in addition to shared-memory architectures.

### Classification

Classification is a technique for the segmentation of medical images. The k-Nearest Neighbor (k-NN) classification rule is a technique for nonparametric supervised pattern classification. Duda, 1973 describes k-NN classification and its properties. Each voxel is labeled with a tissue class selected from a set of possible classes. The possible tissue classes are described, in k-NN

classification, by selecting a set of typical voxels (prototypes) for each tissue type. Voxels of an unknown class are then classified by comparing the voxel intensity characteristics with those of the prototypes and selecting the class that occurs most frequently among the  $k$  nearest prototypes.

The classification of each voxel is independent of neighboring voxels. As such the most straightforward parallelization strategy is to apply the  $k$ -NN classification rule to several voxels at the same time, up to the number of CPUs available for computation. Speedup is linear with the number of CPUs.

### **EM Segmentation**

EM segmentation is a method that iterates between conventional tissue classification and the estimation of intensity inhomogeneity to correct for imaging artifacts. The EM algorithm consists of a conventional classification step, an intensity prediction step, and an intensity correction step. Classification is parallelized by classifying different voxels simultaneously, as above. The same is done with the intensity prediction step. Intensity correction primarily involves low-pass filtering implemented with a parallel unity gain filtering step that costs only two multiplies per voxel per axis, independent of filter length.

## **6.2 Image Processing: Registration**

### **Linear Registration**

Linear registration algorithms align several complementary data sets of the same subject (e.g., a CT and an MRI scan). Another application is the initial alignment, as a preliminary step before non-linear registration, of a canonical data set and the data from a specific subject. Different algorithms that have been published in the literature Warfield, et al 1998, West, et al 1997, typically trade off speed (e.g., through feature extraction or subsampling) and robustness and capture range (e.g., by simulated annealing).

### **Intra-patient Registration**

A common method works with the concept of subsampling of the gray scale data for speed-up. Entropy calculations are performed in a histogram feature space. The algorithm is relatively fast

and does not require any preprocessing of the data. The operator selects three paired landmarks and the algorithm then calculates an alignment to subvoxel accuracy. Alignment is assessed by using inherent contrast similarity to directly measure the image alignment. The algorithm requires entropy and joint entropy computation. Mutual information is defined in terms of entropy. The first term is the entropy in the reference volume. The second term is the entropy of the part of the test volume into which the reference volume projects. It encourages transformations that project the reference volume into complex parts of the test volume. The third term, the (negative) joint entropy of the reference and test volume, contributes when they are functionally related. A histogram-based density estimate is used for the joint entropy estimation. The joint histogram computation is parallelized by dividing data into chunks, computing the histogram of each chunk, and then adding the histograms together. The joint entropy can then be calculated by a loop over the histogram. Acquisitions with different contrasts can be registered into multichannel data sets for better segmentation and visualization. Examples include image analysis in neonates (T2/PD - SPGR,) and surgical planning (MRA, SPECT, fMRI, MRI, CT).

### **Interpatient Registration**

In a situation where it is necessary to align two data sets of different subjects dense feature comparison turns out to be more robust than sparse feature comparison. Parallelization is used to allow the speed-up of dense feature comparisons, making the application of this technique practical in a clinical context.

Segmentations of the patient scans to be aligned are generated; then a measure mismatch of alignment is generated by counting the number of voxels that don't match; then a transform that minimizes the mismatch is determined. Each scan to be registered is classified and a multiresolution pyramid of the classified scan is constructed. An initial alignment is selected as either the identity transform or the transform identified with the process described below. For each level of the pyramid, the optimum alignment is determined by minimizing the mismatch of corresponding tissue labels. Each evaluation of this mismatch can be computed in parallel.

The evaluation of a particular transform involves the comparison of aligned data with a two step process. First the moving data set is resampled into the frame of the stationary data set. Second is the voxelwise comparison of label values. Each of these steps can be parallelized by carrying out the operations simultaneously on some voxels in the frame of the stationary data set. This algorithm was initially developed for interpatient registration such as the initial alignment for template driven segmentation (TDS). TDS is used in many applications such as the quantitative analysis of MS, brain development, schizophrenia, and rheumatoid arthritis. More recently, the algorithm has been used for inpatient alignment, if a large capture range was needed.

### **Non-linear registration**

Local shape differences between data sets can be identified by finding a 3D deformation field that alters the coordinate system of one data set to maximize the similarity of local intensities with the other. Elastic matching aims to match a template, describing the anatomy expected to be present, to a particular patient scan so that the information associated with the template can be projected directly onto the patient scan on a voxel to voxel basis. The template can be an atlas of normal anatomy (deterministic or probabilistic), or it can be a scan from a different modality, or it can be a scan from the same modality. The template can contain information typically found in anatomical textbooks, but unlike normal textbooks, can be linked to any form of relevant digital information.

Algorithmic improvements to speed up the processing include a multiresolution approach with fast local similarity measurement, and a simplified regularization model for the elastic membrane. Algorithms parallelized for SMP such as low pass filter upsampling and downsampling, arithmetic operations, and solving systems of equations are typically used. Nonlinear registration is primarily used for incremental alignment in TDS, following the linear alignment step

## **6.3 Visualization: Surface Model Generation and Volume Rendering**

To visualize the surface of structures by simulating light reflection requires generation of models by segmentation. The process consists of segmentation of the data into binary label maps and application of a surface model generation pipeline consisting of the marching cubes algorithm

for triangle model generation, followed by triangle decimation and triangle smoothing to reduce triangle count. The algorithm is parallelized by distributed computation of triangle models for each structure of a data set. Efficient triangle model generation has been used for the visual verification of segmentation procedures, visualization for surgical planning and navigation.

Visualization of structures without the need for the extensive preprocessing required by the surface model approach can be done using volume rendering. This is of benefit if the structures to be visualized are constantly changing. Ray casting and shear warp algorithms are among the most popular approaches for volume rendering. The algorithm is parallelized by applying the light transmission model simultaneously to different sections of the data associated with different screen pixels.

#### **6.4 Visualization Case Study: National Center for Macromolecular Imaging**

Dr. Wah Chiu is the Alvin Romansky Professor at the Department of Biochemistry & Molecular Biology at the Baylor College of Medicine. He is the director of the National Center for Macromolecular Imaging (NCMI). Together with Dr. Steve Ludtke, Co-Director of NCMI, and Wen Jiang, a post-doctoral researcher, Dr. Chiu discussed his research on the use of electron cryomicroscopy to determine the three-dimensional structures of molecules and macromolecular assemblies at subnanometer resolution. Their laboratory is uniquely equipped with intermediate high-voltage electron cryomicroscopes that have been dedicated to advance the technology for imaging individual macromolecular assemblies embedded in vitreous ice. They are now developing computational procedures to facilitate high throughput and high quality data collection via computer instead of manual control approach. The NCMI has the missions of collaboration, training and dissemination in addition to core research and technology development. Presently, NCMI is engaged with over 50 collaborators/users and 150 projects ranging from solving 3-D structures to exploring grid computing.

Current computational work at NCMI focuses on a technique known as single particle reconstruction. In this technique, electron cryo-microscopy is used to record projection images of individual molecules or macromolecular assemblies embedded in vitreous ice. These randomly oriented particle images are then processed using a complex sequence of algorithms including

multidimensional alignment, deconvolution, Fourier and real-space reconstruction techniques, etc. The final result is a 3D structure of the original molecule. Images are extremely noisy, with typical peak SNR values of 0.6. Digital images range from 64x64 pixels to 1024x1024 pixel arrays. Reconstructions are thus  $64^3$  to  $1024^3$ . For a given macromolecular assembly, between 10,000 and 1,000,000 images are processed to produce the final reconstruction. For a small problem this may take only 100 CPU hours, but typical high-resolution problems require ~10,000 CPU-hours, and the cutting edge problems now being worked on are anticipated to use 100,000-1,000,000 CPU-Hours. The raw image data ranges from 1-100 gigabytes of storage. Memory requirement for typical problems is ~2 Gb/CPU, though larger problems would work better with 8-16 Gb.

Current methods use very coarse-grained parallelism with good scalability to well over 100 CPUs. Larger problems scale better. For this reason, raw price-performance is the main consideration in CPU purchasing decisions. The NCMI currently has a 160 CPU linux cluster dedicated to processing, and this will at least double in the next year. As research moves towards generating higher resolution and larger images requiring longer running jobs, memory bandwidth could become a significant bottleneck. Current algorithms are coarse-grained, but algorithm changes to handle increased image sizes and produce better resolution may require more finely grained code implemented using MPI or other clustering protocol. If MPI-based software is relied on, there are severe fault tolerance problems, since a single node crashing will kill the entire job. More fault-tolerant parallelism would be desirable.

As with the Cardiac Group at UCSD, staff and researchers at NCMI come from a variety of backgrounds and often have little formal computer science training. The ease of use of uncompiled scripting languages has quickened their adaptation in many bio-medical computing environments, including NCMI. Virtually all high-level programming is now done in Python with numerically intensive operations performed by embedded C++ libraries. While Python is not currently used directly for numerically intensive work it would be quite desirable if this were possible in certain cases. The availability of a higher performance python solution is highly desirable, ie – a python compiler or better Just-In-Time (JIT) environment.

Code optimization is a substantial issue. Currently available profiling tools often do not provide fine enough detail to provide significant performance improvements, even when they are possible. The profiler SHARK (Mac OSX) has demonstrated that there are substantial opportunities for optimizations that have been missed in our hand-coding efforts. Better freely available optimization/profiling tools would be very useful. Writing highly optimized code has some drawbacks when it comes to portability, however. It is possible to write code that is relatively platform independent. Experience has shown that it takes about two days worth of effort to port a moderately sized program across platforms. However, this assumes that minimal platform-specific optimizations are performed.

NCMI is has seen extraordinary growth in the number of users taking advantage of their software products and they are preparing for the future. The single particle reconstruction field is growing rapidly. Over the last 2-3 years, the number of researchers using NCMI software (known as EMAN) has nearly tripled to about 300 users. While most biological scientists today are familiar with computers, they may not have the expertise to adapt NCMI software to fit specific research needs. NCMI is seeking to make their technology packaged so that more scientists can use it for their own research. Studies have indicated that there has recently been a drop of nearly 20% of computer science majors in the US ([http://www.usatoday.com/tech/news/2004-08-08-computer-science\\_x.htm](http://www.usatoday.com/tech/news/2004-08-08-computer-science_x.htm)) . If this develops into a trend, manpower issues could be a serious problem as fewer system managers and programmers enter the field although roughly 10% of NIH money is anticipated to be used for bio-computing activities in the coming decade.

## 6.5 Visualization Needs

### Interviewees

#### **Ron Kikinis, M.D.**

Ron Kikinis is the Director of the Surgical Planning Laboratory of the Department of Radiology, Brigham and Women's Hospital and Harvard Medical School, Boston, MA, and an Associate Professor of Radiology at Harvard Medical School, as well as an Adjoint Professor of Biomedical Engineering at Boston University. His interests include the development of clinical applications for image processing, computer vision and interactive rendering methods.

**Joachim Frank, Ph.D.**

Dr. Frank is a Howard Hughes Medical School Institute Investigator. He is also a Professor for the School of Public Health and Biomedical Sciences at the New York State Department of Health, Wadsworth Center. His laboratory develops methods of 3D visualization and structural analysis with the [electron microscope](#), and applies these methods to a number of important biological structures.

**Wah Chiu Ph.D.**

Wah Chiu is the director of the National Center for Macromolecular Imaging (NCMI). Together with Dr. Steve Ludtke, Co-Director of NCMI, and Wen Jiang, a post-doctoral researcher, Dr. Chiu discussed his research on the use of electron cryomicroscopy to determine the three-dimensional structures of molecules and macromolecular assemblies at subnanometer resolution.

**Requirements**

1. The best resolution obtained to date is at 8 Angstroms. At this resolution, alpha helices are visible, but beta sheets are not. At 3 Angstroms resolution, where beta sheets would be visible and most of the structure could be traced, it is estimated that 1 million images would be required. The following capabilities need to be developed to achieve 3 Angstrom resolution:
  - Data collection needs to be automated. Currently, micrographs are collected on film and then scanned. Direct digital readout with sufficiently large (4k x 4k and larger) CCD cameras, and automation of the collection of images are essential for higher productivity.
  - At this resolution, there will be an explosion in the volume of computation. It will be necessary to make better use of existing computational resources, perhaps by developing distributed computing pipelines.
  - Algorithms need to be developed to deal with conformational heterogeneity. An underlying assumption of single particle reconstruction is that the molecules imaged have identical structure. In order to differentiate between possible conformational states, classification algorithms, including self-organized maps and neural networks, may be needed to be folded into existing refinement techniques.

2. Some current bottlenecks in productivity include:

- Scheduling tools use a lot of overhead, so these are not used. This means that the good will of users is required when multiple users are on the system and competition for resources is high.
- Data streams and formats are not homogenous across commercial and academic software. This can present problems for collaborations, but it also prevents academic groups from using multiple software.

3. Algorithmic improvements are handled by specialists. Additional improvements in the reconstruction procedure are expected to come from (i) the inclusion of classification algorithms, (ii) a refinement scheme that is based on a simultaneous solution of the reconstruction and alignment problem, and (iii) updating algorithms as platforms change to take advantage of improved performance offered by new platforms.

4. Digital images range from 64x64 pixels to 1024x1024 pixel arrays. Reconstructions are thus  $64^3$  to  $1024^3$ . For a given macromolecular assembly, between 10,000 and 1,000,000 images are processed to produce the final reconstruction. For a small problem this may take only 100 CPU hours, but typical high-resolution problems require ~10,000 CPU-hours, and the cutting edge problems now being worked on are anticipated to use 100,000-1,000,000 CPU-Hours. The raw image data ranges from 1-100 gigabytes of storage. Memory requirement for typical problems is ~2 Gb/CPU, though larger problems would work better with 8-16 Gb.

5. As research moves towards generating higher resolution and larger images requiring longer running jobs, memory bandwidth could become a significant bottleneck. Current algorithms are coarse-grained, but algorithm changes to handle increased image sizes and produce better resolution may require more finely grained code implemented using MPI or other clustering protocol.

6. Desirable productivity tools that address the following issues:

- Code optimization is a substantial issue. Currently available profiling tools often do not provide fine enough detail to provide significant performance improvements, even when they are possible. The profiler SHARK (Mac OSX) has demonstrated that there are substantial opportunities for optimizations that have been missed in our

hand-coding efforts. Better freely available optimization/profiling tools would be very useful.

- If MPI-based software is relied on, there are severe fault tolerance problems, since a single node crashing will kill the entire job. More fault-tolerant parallelism would be desirable.

- Virtually all high-level programming is now done in Python with numerically intensive operations performed by embedded C++ libraries. The availability of a higher performance python solution is highly desirable, i.e. – a python compiler or better JIT environment.

- Experience has shown that it takes about two days worth of effort to port a moderately sized program across platforms. However, this assumes that minimal platform-specific optimizations are performed.

7. Providing the surgeon with real-time information about the effects of surgical decisions is a substantial computational burden on computing systems. Data augmentation algorithms must accurately predict the response of tissue to surgical invasion. During brain surgery, finite element mesh simulations, with millions of elements, can saturate a peta-flop machine.

8. Providing image guided therapy depends critically on memory bandwidth and latency; performance is relatively independent of architecture.

Visualization in biological and medical research has rapidly emerged as a unique and significant discipline aimed at developing approaches and tools to allow researchers to "see into" and comprehend the living systems they are studying. Topics of investigation and development in the discipline span from basic theory through tools and systems to complete applications. The benefits of medical imaging systems have clearly been established in several areas, including improved training, better diagnosis, and accuracy in performing certain conventional surgical procedures. Ultimately, the continual improvement of visualization technology, including developments in algorithmic, software, and computer processing, will pave the way for its permanent integration into surgery, healthcare delivery, and medical education.

## Chapter 7: Virtual Soldier Project

The [DARPA](http://www.virtualsoldier.net) Virtual Soldier Project ([www.virtual soldier.net](http://www.virtualsoldier.net)) is investigating methods to provide improved medical care for the soldier. The project is focused on production of complex mathematical models to create holomers, or physiological representations of individual soldiers that can be used to improve medical diagnosis on and off the battlefield. The holomer data will be coupled with predictive modeling software, to facilitate a new level of integration in medical procedures. The Virtual Soldier will provide multiple capabilities, including automatic diagnosis of battlefield injuries; prediction of soldier performance; evaluation of non-lethal weapons; and virtual clinical trials..

### 7.1 The Use of Hydrocode and Other Modeling Approaches for Modeling the Ballistic Wounding of Tissue

#### **Robert Eisler, Mission Research Corporation**

Hydrocode, developed at the Lawrence Livermore Laboratory, is often used for modeling ballistic interactions and what happens to projectiles under high energy transfer settings. For this reason the study of this phenomenon is known as Hydrocode analysis (hydra due to liquid, code is due to the requirement for sufficient computer power for modeling and simulation).

Hydrocodes's are a popular field of applied mathematics and are widely recognized as a valuable resource in mathematical modeling under high impact situations, where huge energy transfers occur. Hydrocodes were developed because under high temperatures and pressures, many robust materials, such as ceramics and iron, behave more like a liquid than a solid. However, this approach may be limited because biological tissue already contains a great deal of water, and is structurally much more complex than uniformly hard materials such as iron or ceramics. In addition, little is actually known about the material properties of tissue, although some resources exist (see. Thus, more work in this domain may be useful, and if the computing power is available, and the research can be more focused on integration of accurate biological models into the studies, then more success may be anticipated – one of the goals of the Virtual Soldier Project and other efforts such as the Digital Human project.

## Specific Approach

The role that Robert Eisler and the Mission Research Corporation plays in the Virtual Soldier Project could benefit greatly from HPCS. Their goal is to develop analytic models that describe the tissue damage from ballistic experiments that produce penetrating wounds to the heart in an animal model. The complexity of this approach is staggering, and reasonable timeframes require massive computing power.

Acoustic measurements are taken from the porcine tissue that is traversed by the wound tract and they will be used to develop stress-strain models for the tissue material as a function of frequency. However, more detailed tissue models are forthcoming, adding exponential complexity to this problem.

The Lamé constants,  $\lambda$  and  $\mu$ , arise in stress-strain relationships and can be used to express other solid material properties such as Young's modulus, Poisson's ratio, Bulk modulus, and Shear modulus. To derive the Lamé constants, the group will use the relationships between the constants, density and sound speed measured for each given solid sample. These relationships are given in the following equations:

$$c_L = \sqrt{(\lambda + 2\mu)/\rho}$$

$$c_s = \sqrt{\mu/\rho}$$

where  $c_L$  = the speed of sound for a longitudinal wave,

$c_s$  = the speed of sound for a shear wave

$\rho$  = density

both  $c_s$  and  $c_L$  are a function of the frequency of the sound wave.

After mapping material properties to a porcine anatomical model, wound trajectories can be simulated. It will be assumed that projectiles suffer only minor deformations in the wound tract so that rigid body dynamics can be used. Determining the wound tract consists of determining the motion of the center of mass of the projectile and the motion of the projectile about its center of mass (i.e. rotations and small deformations perhaps). Experiments have been carried out by shooting small spheres and other projectiles into gelatin targets to obtain the penetration depth as

a function of initial velocity given some type of material. By analytically inverting the data yielded by these experiments, it is possible to solve for the retardation force as a function of instantaneous projectile velocity.

To do this, the problem can be first reduced to finding the retardation force  $F(v)$  per unit mass on a spherical projectile, given empirical data relating the distribution of the penetration depth,  $\delta(v_0)$ , to the entry velocity  $v_0$ . It is also assumed that at low velocities, the retardation force becomes constant, and at high velocities (near the speed of sound), the retardation force is proportional to the square of the instantaneous velocity.

The relationship between delta and  $F(v)$  is:  $F(v) = v/(d\delta/dv)$ . For low velocities, the proposal taylor expands the function delta as  $\delta(v) = v^2(a + bv + cv^2)$ . For high velocities,  $\delta(v) = a' \ln(v) + b'/v + c'/v^2$ . The data can be then fitted to these equations to yield the coefficients for delta. Delta can then be differentiated with respect to  $v$  and substituted into the equation for  $F(v)$ .

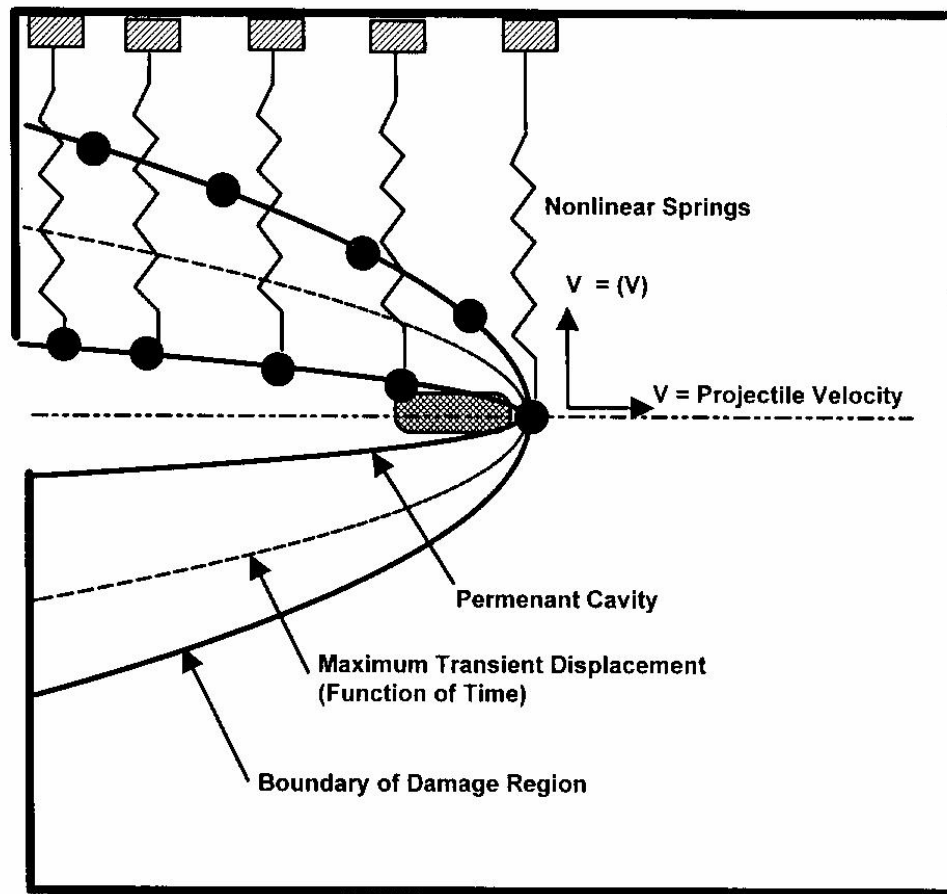


Figure 7.1: Parallelization of Material Point Method (MPM) Codes for Torso and Cardiovascular Wounds

Christopher Johnson, Ph.D., University of Utah

### Material Point Method (MPM)

MPM is a new particle-based method for simulations in computational solid and fluid mechanics. This method uses a regular structured grid as a computational scratchpad for computing spatial gradients of field variables for representation in the equations of motion (Figure 1A, green lines). The materials of interest are discretized with particles rather than a traditional connected mesh. The grid is convected with the particles (Figure 1B red circles) during deformations that occur over a time-step, eliminating the diffusion problems associated with advection on an Eulerian grid. The grid is restored to its original configuration at the end of a time-step while the particles remain deformed/convected (Figure 1C). Coupled solid-fluid mechanical simulations are readily performed with MPM because a regular grid is used for

gradient calculations. The grid then serves as both an Eulerian reference frame for CFD calculations and an updated Lagrangian reference frame for MPM calculations. MPM has several significant advantages for the presently proposed research. In particular MPM makes it possible to simulate the evolution of complex free surfaces and the fragmentation of those surfaces. The method has been used with great success to model the fragmentation of solid bodies as part of the Department of Energy sponsored Center for the Simulation of Accidental Fires and Explosions (C-SAFE) at the University of Utah. The particular strength of MPM is that it avoids issues of element inversion and mesh entanglement during the simulations and is straightforward to parallelize.

MPM codes have been parallelized in Utah using Message Passing Interface (MPI) for execution on distributed memory machines [1], which will allow the simulation of extremely large problems. The explicit MPM code was implemented within the Uintah Computational Framework (UCF)<sup>80</sup> and contains a very high content of parallel code. This framework was developed at the University of Utah, initially in support of the Department of Energy sponsored Center for the Simulation of Accidental Fires and Explosions (C-SAFE), and utilizes domain decomposition to achieve parallelization. Previous implementations of the MPM algorithms have utilized explicit time integration. This approach will be especially useful in further work in simulating events such as a slow moving bullet fragment or shrapnel impacting the heart.

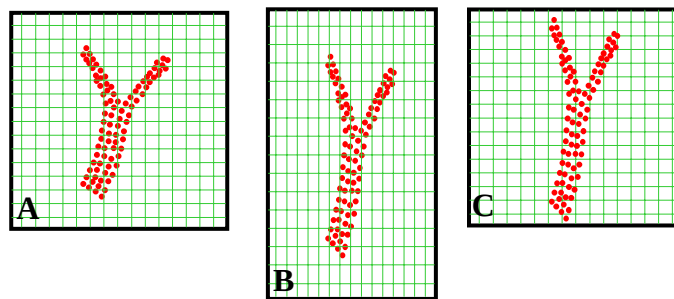


**Figure 7.2: Schematic of a single computational step in MPM algorithm.**

A) Initial distribution of particles (red) and background computational grid (green). B) Stretching (vertical) and contraction (lateral) applied to particles. Computational grid convects with particles. C) Computational grid is reset and particles remain deformed/convected.

simulations of torso wounds that will use computational models of the cardiovascular system created at Auckland, UCSD and Utah. The challenge of computing deformation and structural

damage is being addressed collaboratively with the researchers at UCSD, with the Utah team pursuing the use of both conventional finite element techniques and the material point method (MPM).

The latter method allows substantial structural changes to be readily computed through the explicit movement of tissue across the mesh. Experience with MPM methods has shown that it provides accurate solutions for problems involving large deformations such as projectile intrusions, without the drawback of mesh distortion that is typically encountered in the finite element method<sup>81</sup> [2].

A key research issue is that of resetting the computational grid at the end of each time-step. At each time-step, a displacement is computed at every computational node. In principle, the grid has deformed according to these displacements. In practice, these displacements are interpolated to the particles, which are then displaced accordingly, in essence moving them through the grid. This is a sensible choice for simulations that involve deformations that may lead to mesh entanglement. The initial grids used will be the three-dimensional Hermite cubic meshes developed by San Diego. This aim involves mapping the MPM solutions onto the meshes needed by the other simulation components. An investigation of this will be a key research task in the first part of the project. It will also be of great importance to compare this approach with the more standard finite element approach used by the San Diego group. A preliminary assessment of MPM usefulness with regard to the problem area will be a key deliverable of the project. A similar assessment<sup>80</sup> for a different problem class has already provided encouragement.

## Chapter 8: Interviews

We conducted interviews with researchers in several fields at various professional levels, from graduate students to post docs to researchers to directors. In total we conducted interviews with representatives from 15 groups. The individuals had positions at various career levels within the computational community, from directors of large centers to graduate students.

### 8.1 Stanley K. Burt, Ph.D.

Director of the National Cancer Institute's Advanced Biomedical Computing Center (ABCC).

Phone: 301-846-2178

Email: [burt@ncifcrf.gov](mailto:burt@ncifcrf.gov)

<http://www-fbnc.ncifcrf.gov/>

Stan Burt, Ph.D. is the Director of the National Cancer Institute's Advanced Biomedical Computing Center (ABCC). The National Cancer Institute's supercomputing facility is a fully integrated, high performance, scientific computing resource located at the [NCI-Frederick](#) campus in Frederick, MD. The facility provides state-of-the-art computing support and technology to the scientists of the [National Cancer Institute \(NCI\)](#), [National Institutes of Health \(NIH\)](#), and extramural biomedical researchers. Unlike some other high performance centers, ABCC only supports research directed toward biological problems, thus making it somewhat unique. For more information, see the presentation available at: [www.fas.org/biomed\\_hpcs](http://www.fas.org/biomed_hpcs).

#### Summary of key points

The key ideas conveyed at the interview with Dr. Burt can be summarized as follows:

1. ABCC is focusing on grand challenge type problems such as deciphering the human genome, understanding structure-function relations of biological molecules, and development of therapeutics for numerous diseases, especially AIDS and cancer.
2. The center provides a heterogeneous computer environment in order to let researchers match their specific problem needs to the appropriate platform. The center maintains suite of programs ranging from bioinformatics, to molecular dynamics, to quantum chemistry.

3. Quantum chemistry has not been heavily exploited for studying biological systems mainly due to limitations in computer power and memory.
4. Computers are only now attaining speeds and memory size that allow quantum investigations of biological systems to be fruitful - but, even so, the size of the problem that can be studied is still severely limited and there is the need for new algorithmic development.
5. Ab initio and density functional methods (DFT) offer the best hope for understanding enzyme mechanisms, hydrogen bonding, polarization effects, spectra, Van der Waals interactions and other fundamental processes in biology.
6. Theoretical approaches that assess ligand-protein binding affinity prior to synthesis and testing of ligands are of obvious importance in the field of structure based drug design. A correlated ab initio quantum mechanical treatment using on the order of 200 or more active site atoms with large basis sets containing polarization and diffuse functions is necessary for an accurate characterization of ligand-binding processes. The iterative solution of the Poisson equation to determine polarization requires a solution of 80,000 coupled equations.
7. Analysis has shown that in order to obtain absolute binding free energies to within one kcal/mol precision it will be necessary to solve 500,000 to one million coupled equations. In order to accomplish this, a one or two order of magnitude increase in computational power will be required.

## **8.2 Brett Peterson, Ph.D.**

NIH-NCRR, Health Science Administration

301.435.0758

[petersob@mail.nih.gov](mailto:petersob@mail.nih.gov)

Dr. Peterson is health scientist administrator in NCRR's Division of Biomedical Technology. The interview was held at the January 2003 workshop. He discussed the Biomedical Informatics Research Network (BIRN), a [National Institutes of Health](#) initiative that fosters distributed collaborations in biomedical science by utilizing information technology innovations. Currently the BIRN involves a consortium of 14 [universities](#) and 22 [research groups](#) that participate in one or more of three test bed projects centered around brain imaging of human neurological disorders

and associated animal models. For more information, see the presentation available at: [www.fas.org/biomed\\_hpcs](http://www.fas.org/biomed_hpcs).

### **Summary of key points**

The most important key ideas conveyed at the interview with Dr. Peterson, NIH-NCRR, Health Science Administration, can be summarized as follows:

1. BIRN, started in 2001, links together some of the Nation's top brain-imaging research centers, allowing scientists to share, compare, and morph together brain scans from healthy individuals, patients with brain disorders, and nonhuman species, such as mice with neurological conditions.
2. Goals are to: establish distributed and linked data collections for investigators' research projects; enable access to heterogeneous "grid-based" computing resources for research project analyses; provide data mining tools to search multiple data collections or databases; develop the software and hardware infrastructure that will allow scientists to conduct valid multisite neuroimaging studies, for example.
3. BIRN will build three federated databases: Morphometry BIRN, which focuses on human brain structure; Function BIRN, which analyzes functioning of the human brain; and Mouse BIRN, which emphasizes studies of the mouse brain and cross-species comparisons.
4. Morphology BIRN is focuses on combining data from multiple acquisition sites and increasing the statistical power for studying relatively rare populations.
5. Functional BIRN is developing a common fMRI protocol to study regional brain dysfunction related to the progress and treatment of schizophrenia and investigating techniques to insure interoperability of existing tools for multi-model analysis.
6. Mouse BIRN is studying animal disease models across dimensional scales to test hypothesis with human neurological disorders.
7. The BIRN coordinating center will build a fully integrated architecture on top of the network for sharing and mining data (the broadband Abilene and Calren-2 network backbones serve as the foundation for the BIRN network.) Globus and the Storage Resource Broker (SRB) are examples of services and applications that will be used. On top of the data and computational grid, data modeling and integration tools will be incorporated and developed to facilitate the

construction of project- and site-specific data models; extend methods to query and retrieve complex data and associations from each partnering site; and develop a body of "domain knowledge" to allow correlation of data in the BIRN archives. A universal visualization toolkit will be developed, incorporating elements and tools from each of the partnering site along with comprehensive file format converters. The Grid Security Infrastructure (GSI) system within the Globus toolkit will provide services to address security, including authentication, encryption, and enforcement of a signed certificate authority. For the Brain Morphology scientific project, human data will be appropriately sanitized prior to insertion into the BIRN file system as appropriate to satisfy HIPPA requirements.

### **8.3 Giri Chukkapalli, Ph.D.**

Scientific Computing Group  
San Diego Supercomputer Center

Dr. Giri Chukkapalli received his PH.D in Mechanical Engineering at the University of Toronto, focusing his dissertation on developing weather models on parallel computers. He is an assistant programmer/analyst at SDSC, where he is involved with several projects, including code (MPI) parallelization, the IBM S/390 supercomputer, and research involving computational fluid dynamics. He discussed Protein Structure Predication using Structure Fragment Libraries work he is doing with Prof. Shankar Subramanyam. For more information, see the presentation available at: [www.fas.org/biomed\\_hpcs](http://www.fas.org/biomed_hpcs).

#### **Summary of key points**

1. The research team with which he works uses an ab initio MD based approach. It is compute intensive, data intensive, irregular and pipeline driven.
2. Their procedure is as follows:

- Generation of a comprehensive fragment library
- Clustering fragments to generate the structural alphabet
- Matching target sequence fragments to the structure fragments
- Stitching of fragments to generate a model structure
- Global and local optimization of model structure

Sanity check

3. The fragment library is written in Oracle database and is approximately 20k lines of code.
4. The clustering algorithms are computationally intensive and are currently implemented on a SUN E15k with 72 processors and 300 GB of memory, using a java thread library. The algorithms are easier to implement using shared memory parallelism due to tight coupling, fine grained, non-uniform work load
5. The model structure optimization uses a GA algorithm and is compute intensive.
6. Requirements identified are:
  - a. Hardware/software to support the pipeline processing efficiently (other fields have similar needs, including climate modeling)
  - b. Tools to schedule such a pipeline and for checkpointing
  - c. Well-balanced hardware pipeline from archival storage to compute elements without bottlenecks
  - d. Easily programmable FPGA coprocessor boards to handle integer and other DSP branch of the pipeline
  - e. Hardware and software to handle the overlapped computation, communication and I/O
  - f. Efficient ANN and GA libraries similar to LAPACK

## **8.4 Joel Stiles, Ph.D.**

Associate Professor, Mellon College of Science & Pittsburgh Supercomputing Center

Biomedical Applications, Pittsburgh Supercomputing Center

[stiles@psc.edu](mailto:stiles@psc.edu)

Home page: <http://www.psc.edu/~stiles/>

Phone: 412 - 268-4786

Dr. Joel Stiles is Associate Professor, Mellon College of Science. He and Tom Bartol (Computation Neurobiology Laboratory, Salk Institute) developed MCell, a simulation program that makes it possible to incorporate high resolution ultrastructure into models of ligand diffusion and signaling: MCell is a general Monte Carlo simulator of cellular microphysiology. Diffusion

of individual ligand molecules is simulated using a Brownian dynamics random walk algorithm, and bulk solution rate constants are converted into Monte Carlo probabilities so that the diffusing ligands can undergo stochastic chemical interactions with individual binding sites such as receptor proteins, enzymes, transporters, etc.

### **Summary of key points**

1. Until recently, computational limits have precluded highly realistic 3D simulations of subcellular architecture and physiology. As a result, the contribution of actual ultrastructure to signaling variability and plasticity has gone largely unexplored, and quantitative modeling has been severely hampered.
2. MCell's use to date has been focused on one aspect of biological signal transduction, namely the microphysiology of synaptic transmission, but other areas of possible application include statistical chemistry, diffusion theory, single channel simulation and data analysis, noise analysis, and Markov processes.
3. Biological structures show tremendous complexity and diversity at the subcellular level. For example, a single cubic millimeter of cerebral cortex may contain of order 5 billion interdigitated synapses of different shapes and sizes.
4. MCell is being parallelized with support from National Partnership for Advanced Computational Infrastructure (NPACI), to allow a single MCell job to split the computational load of its diffusion algorithm and its large memory requirements.
5. Parallelization is challenging because although there is a maximum diffusion within a time step (and there are hundreds of thousands of time steps), it is not possible to know how many molecules will enter from nearby processors. Processor boundaries introduce uncertainties in handling communications and the need to detect termination.
6. Single large-scale simulations on massively parallel supercomputers is accomplished using MCell with KeLP, called MCell-K. On the NPACI Blue Horizon system, a 256 processor version runs at approx. 70%.
7. An example simulation to map part of a 4-D parameter space representing the transmission behavior at a nerve-muscle synapse required 47,040 runs, which was completed in 48 hours running on a combination of 512 and 1024 processors and generated 50 gigabytes of output data representing new disciplinary results.

8. Key requirements: random number generators; vastly improved reliability of systems; tools to support load balancing.

## 8.6 Brian Athey, Ph.D.

Director, Michigan Center for Biological Information (MCBI)

Assistant Professor Biomedical Informatics

Director, University of Michigan Visible Human Project

bleu@umich.edu

Dr. Athey is Director of the Michigan Center for Biological Information at the University of Michigan . MCBI provides advanced bioinformatics and computational resources for investigators in the academic and industrial sectors of Michigan. Researchers will have access to bioinformatics tools, genomics and proteomics databases, supercomputing resources, bioinformatics training, and bioinformatics consulting through MCBI. MCBI is researching appropriate hardware, middleware, and networking structures for state-wide analysis and data-sharing in bioinformatics projects. For more information, see the presentation available at: [www.fas.org/biomed\\_hpcs](http://www.fas.org/biomed_hpcs).

### Summary of key points

1. General needs in biomedical sciences can be enabled by next generation supercomputing, for example: mouse/human genome correlation; individual pharmacogenomic analysis using gene expression arrays; multi-modal radiology image fusion; millisecond structural biology enabled by synchrotron x-ray sources and 900 Mhz NMR; physiologically competent Digital Human Simulations
2. Not all biology problems are embarrassingly parallel; shared memory with database(s) close in is preferred in many (most) biologically interesting problems.
3. As data sizes grow data motion will bottleneck the computing progress. For example, in the 100 seconds it takes to move the 1 GB file, a 1 GHz machine could perform 0.1TeraOP. Data needs to be local, and stay local. Data motion needs to be asynchronous, and happen at near wire speeds. Throwing money at the network does not solve the problem.

4. Data size issues dominate: processing time, data motion and data storage. Data must be distributed, Data I/O must occur over many channels, and have no single points of flow.
5. Most bioinformatic applications are: integer bound; memory latency bound, and pointer chasers (cache thrashers).
6. Key informational needs of researchers:
  - Almanac or index that would link every human gene to all the information known about these genes from the literature, from all relevant expenditures and other sources.
  - Better Relational databases helping researchers to move from a gene by gene approach
  - More focus on patterns and pattern recognition
  - Better and system wide in silico models of human

## **8.7 Juan Cebral, Ph.D.**

Assistant Research Scientist

George Mason University

Dr. Juan Cebral discussed the research he and the research team including Rainald Loehner, and Orlando Soto, George Mason Univ.; and Peter L. Choyke and Peter J. Yim, National Institutes of Health. The application he discussed is an image-based finite element model of hemodynamics in stenose carotid, a methodology to construct patient-specific, anatomically and physiologically realistic finite element models of blood flows. Their approach uses MRA data to obtain all the anatomical and physiologic data necessary for realistic modeling of blood flows in carotid arteries with stenosis. The application has the potential for use to characterize healthy and diseased flow and wall shear stress patterns. For more information, see the presentation available at: [www.fas.org/biomed\\_hpcs](http://www.fas.org/biomed_hpcs).

### **Summary of key points**

1. Anatomical models of carotid arteries with stenosis are reconstructed from contrast-enhanced magnetic resonance angiography (MRA) images using a tubular deformable model along each arterial branch.

2. A surface-merging algorithm is used to create a watertight model of the carotid bifurcation for subsequent finite element grid generation.
3. A fully implicit scheme is used to solve the incompressible Navier-Stokes equations on unstructured grids in three-dimensions.
3. Physiologic boundary conditions are derived from cine phase-contrast MRA flow velocity measurements at two locations below and above the bifurcation.
4. The methodology was tested on image data of a patient with carotid artery stenosis. A finite element grid was successfully generated from contrast-enhanced MRA images, and pulsatile blood flow visualizations were produced. Visualizations of the wall shear stress distribution and of changes in both its magnitude and direction were produced.
5. These capabilities may be used to advance understanding of the generation and progression of vascular disease, and may eventually allow physicians to enhance current image-based diagnosis, and to predict and evaluate the outcome of interventional procedures non-invasively.
6. Potential other applications include: study the role of the communicating arteries during arterial occlusions and after endovascular interventions, calculate transport of drugs, evaluate accuracy of 1D flow models, and evaluate vascular bed models used to impose boundary conditions when flow data is unavailable or incomplete.

## **8.8 Ron Kikinis, M.D.**

Surgical Planning Laboratory

Radiology; ASBI, L1-050

Brigham & Women's Hospital

75 Francis St.

Boston, MA 02115

**[kikinis@bwh.harvard.edu](mailto:kikinis@bwh.harvard.edu)**

Home page: <http://www.splweb.bwh.harvard.edu:8000/pages/pp1/kikinis/>

Phone: (617) 732-7389

[Dr. Kikinis](#) is the Director of the Surgical Planning Laboratory of the Department of Radiology, Brigham and Women's Hospital and Harvard Medical School, Boston, MA, and an Associate Professor of Radiology at Harvard Medical School, as well as an Adjoint Professor of

Biomedical Engineering at Boston University. His interests include the development of clinical applications for image processing, computer vision and interactive rendering methods. He is currently concentrating on developing fully automated segmentation methods and introducing computer graphics into the operating room.

### **Summary of Key Points**

1. There are two general types of computation at the Surgical Planning Institute: data-mining and integrated modeling/simulation. Data-mining tasks are performed with the using a pipeline process to handle MRI and CT images in a pooled environment suitable for grid computation. Integrated modeling/simulation involves augmenting images with the results of models and simulations constrained by sparse data acquisition.
2. During surgery the patient changes and the patient's pre-operative imagery is out-dated. Providing the surgeon with real-time information about the effects of surgical decisions is a substantial computational burden on computing systems. Data augmentation algorithms must accurately predict the response of tissue to surgical invasion. During brain surgery, finite element mesh simulations, with millions of elements, can saturate a peta-flop machine.
3. Providing image guided therapy depends critically on memory bandwidth and latency; performance is relatively independent of architecture.
4. As research horizons expand, new inefficiencies both from algorithms and computer power are revealed. This is due, in part, to a lack of knowledge at the boundaries of a field of scientific pursuit.

## **8.9 The Cardiac Group at UCSD**

Dr. Taras Usyk

Sarah Healy

Cardiac Mechanics Group

University of California, San Diego

Phone: (858)822-0346

Web: <http://cmrg.ucsd.edu/>

Dr. Taras Usyk is an Assistant Project Scientist and Sarah Healy is a 3<sup>rd</sup> year graduate student. Both are researchers in the Cardiac Mechanics Group directed by Andrew McCulloch. [Andrew McCulloch](#) is Professor of Bioengineering at the [University of California San Diego](#). He is also a member of the [Whitaker Institute of Biomedical Engineering](#), the UCSD/Salk Institute for Molecular Medicine and the Center for Research in Biological Structure, a Senior Fellow of the [San Diego Supercomputer Center](#), and Director of the [BioNOME Resource](#) at the San Diego Supercomputer Center. Dr. Usyk and Sarah Healy collaborate on structurally and functionally integrated numerical models of cardiac electromechanics using the finite element method. Their goal is to perform whole organ simulations with biophysically detailed systems models involving over a hundred thousand degrees of freedom.

### **Summary of key points:**

1. Electrophysiological models of the heart require small temporal and spatial scales, determined by the system of ODEs that comprise the cellular model. An operator splitting algorithm allows ODE and PDE systems to be solved separately with updates occurring every half timestep. Of these, the ODEs dominate with 90-95% of compute time being spend on them. However, this system is also data parallel and is where they are focusing their efforts.
2. An example of an electrophysiological model that incorporates a mere  $2 \times 1 \times 0.5 \text{ cm}^3$  volume of the heart ventricle uses 1024 finite elements in a tricubic spline approximation; state variables are evaluated at over 10,000 points in the volume. This represents  $1/24^{\text{th}}$  of the whole rabbit ventricle. The code currently requires 300 Mbytes of memory to run; about 250 Mbytes are used by the linear solver and the rest for the input file. Memory requirements scale roughly linearly with the mesh size. The code is written in a combination of python and Fortran, with the use of a linear solver written in C. It currently takes 10 hours to solve the wedge mesh to 1 second, with a time step of 0.1 ms on a single processor Pentium 4 running Linux. We would expect close to linear speedup on a parallel machine.
3. The mesh size for mechanical models of the heart is determined by the ability of solvers to converge to stable solutions. While mechanical models typically require a fraction of the number of mesh points that electrophysiological models require, they have larger

memory requirements, requiring 4-5 Gbytes. This presents a problem when running on machines that have 2-4 Gbytes of memory; it often means simplifying components of the model.

4. Embedding non-native code into a code base, for example embedding Fortran into Python, introduces technical issues with latency and compile efficiency. There is latency associated with handoff of I/O operations. Native and non-native code has to be compiled and debugged separately.
5. There is a need for better debuggers and standard option on compilers. Relying on “print” statements for debugging is time consuming.
6. O/S stability is an issue. Some versions of linux are more likely to crash than others. Productivity could be increased with better strategies for dealing with multiple users vying for the same memory. Currently, users are simply kicked off when too many users are on client/server systems. This results in loss of data and time.
7. The demand for more computing power will never saturate. Currently, heart simulations run for only several to a couple of heart beats. Increased computational power will allow researchers to run to simulate minutes instead of seconds, using more accurate models and including pathologies and cellular level information to gain more robust information about heart function.

### **8.10 Adam P. Arkin, Ph.D.**

Faculty Scientist, Physical Biosciences LBNL.

Assistant Professor, Bioengineering, University of California Berkeley.

Assistant Investigator Howard Hughes Medical Institute

**aparkin@lbl.gov**

Home page: <http://www.http://gobi.lbl.gov/~aparkin/>

Phone: 510-495-2366

Dr. Adam Arkin is an Assistant Professor of Bioengineering at the University of California, Berkeley. He is also a Faculty Scientist in Physical Biosciences at the Lawrence Berkeley National Laboratory. He is one of the central developers of BioSPICE and the director of the Virtual Institute of Microbial Stress and Survival (<http://vimss.org>). BioSpice researchers include biologists, computer scientists, engineers, mathematicians, and software developers. Their

collaborative research programs include experimental and computational studies of *Bacillus subtilis*, development of biophysical theory and software tools for analysis of the dynamics of evolutionary and cellular processes, and development of software tools to graphically represent cellular pathways, define conceptual and mathematical models of processes in the pathway, and link databases to biological entities in the pathway.

### **Summary of Key Points**

1. There are two areas that are benefiting from HPC systems: spatial modeling and simulation of reaction-diffusion equations (algorithms do not scale well), and stochastic estimation and simulation (the latter being embarrassingly parallel).
2. Spatial modeling of cellular processes, for example immune cell chemotaxis, sometimes requires mixed simulation modes that combine models of reaction-diffusion equations for the signaling network with discrete mechanical operations for the cytomachanical processes that change the cell shape in response to chemical gradient signals processed by the signal transduction network. The mechanical and chemical processes are coupled in a feedback loop and thus cannot be separated.
3. Cellular processes take place on many time scales; different reactions have different characteristic rates. Including cellular mechanics introduces another set of times scales. Algorithm development and formal abstraction will probably be the most important aspect of dealing with simulations involving multiple time scales. New algorithms should both be able to separate slow and fast time scales, with well-understood and defined approximation errors, and be numerically stable.
4. An example simulation of a reaction-diffusion equation in 3-D involves 500 state variables (chemical species) and on the order of 500 equations. Including parameter estimation and sensitivity analyses makes this simulation computationally demanding.
5. Simulations of the stochastic dynamics of gene expression and other biochemical processes have 10-15 state variables. These run time for these simulates depends exponentially on the reaction rates, with faster reactions requiring more run time.

6. Both spatial simulations and stochastic simulations can easily require 2 gigs of memory.
7. Code diagnostic tools need to be updated. Code has become sufficiently complicated that a real tool to for designers and project managers to visualize large coding projects would improve productivity.
8. I/O operations tend to be slow. While data files may only be on the order of a few gigabytes, transferring data around networks can be cause bottlenecks.
9. I/O operations tend to be slow. While data files may only be on the order of a few gigabytes, transferring data around networks can be cause bottlenecks.
10. Communication among participants in large modeling and code development projects can be difficult and lead to delays.
11. A considerable amount of time is spent compiling code and it is often easier to reproduce code rather than re-use code written by another researcher using a different compiler. Compile time could be reduced drastically by designing compilers that are efficient across platforms but compile code from anywhere without library dependencies or with very well defined and packaged library dependencies.
12. Better infrastructure that includes rational standards for documentation, data representation, and code needs to be developed.
13. While there have been several attempts, there is still no good visualization tools for large scale, high dimensional data sets.
14. Code diagnostic tools need to be updated. Code has become sufficiently complicated that a real tool to for designers and project managers to visualize large coding projects would improve productivity.

### **8.11 Wah Chiu Ph.D.**

Department of Biochemistry & Molecular Biology

Baylor College of Medicine

wah@bcm.tmc.edu

Home page: <http://ncmi.bcm.tmc.edu/ncmi>

Phone: (713) 798-6985

Dr. Wah Chiu is the Alvin Romansky Professor at the Department of Biochemistry & Molecular Biology at the Baylor College of Medicine. He is the director of the National Center for Macromolecular Imaging (NCMI). Together with Dr. Steve Ludtke, Co-Director of NCMI, and Wen Jiang, a post-doctoral researcher, Dr. Chiu discussed his research on the use of electron cryomicroscopy to determine the three-dimensional structures of molecules and macromolecular assemblies at subnanometer resolution. Their laboratory is uniquely equipped with intermediate high-voltage electron cryomicroscopes that have been dedicated to advance the technology for imaging individual macromolecular assemblies embedded in vitreous ice. They are now developing computational procedures to facilitate high throughput and high quality data collection via computer instead of manual control approach. The NCMI has the missions of collaboration, training and dissemination in addition to core research and technology development. Presently, we are engaged with over 50 collaborators/users and 150 projects ranging from solving 3-D structures to exploring grid computing.

### **Summary of key points**

1. Current computational work focuses on a technique known as single particle reconstruction. In this technique, electron cryo-microscopy is used to record projection images of individual molecules or macromolecular assemblies embedded in vitreous ice. These randomly oriented particle images are then processed using a complex sequence of algorithms including multidimensional alignment, deconvolution, Fourier and real-space reconstruction techniques, etc. The final result is a 3D structure of the original molecule. Images are extremely noisy, with typical peak SNR values of 0.6.
2. Digital images range from 64x64 pixels to 1024x1024 pixel arrays. Reconstructions are thus  $64^3$  to  $1024^3$ . For a given macromolecular assembly, between 10,000 and 1,000,000 images are processed to produce the final reconstruction. For a small problem this may take only 100 CPU hours, but typical high-resolution problems require ~10,000 CPU-hours, and the cutting edge problems now being worked on are anticipated to use 100,000-1,000,000 CPU-Hours. The raw image data ranges from 1-100 gigabytes of storage. Memory requirement for typical problems is ~2 Gb/CPU, though larger problems would work better with 8-16 Gb.

3. Current methods use very coarse-grained parallelism with good scalability to well over 100 CPUs. Larger problems scale better. For this reason, raw price-performance is the main consideration in cpu purchasing decisions. The NCMI currently has a 160 CPU linux cluster dedicated to processing, and this will at least double in the next year.
4. As research moves towards generating higher resolution and larger images requiring longer running jobs, memory bandwidth could become a significant bottleneck. Current algorithms are coarse-grained, but algorithm changes to handle increased image sizes and produce better resolution may require more finely grained code implemented using MPI or other clustering protocol.
5. The single particle reconstruction field is growing rapidly. Over the last 2-3 years, the number of researchers using NCMI software (known as EMAN) has nearly tripled to about 300 users. While most biological scientists today are familiar with computers, they may not have the expertise to adapt NCMI software to fit specific research needs. NCMI is seeking to make their technology packaged so that more scientists can use it for their own research.
6. Studies have indicated that there has recently been a drop of nearly 20% of computer science majors in the US ([http://www.usatoday.com/tech/news/2004-08-08-computer-science\\_x.htm](http://www.usatoday.com/tech/news/2004-08-08-computer-science_x.htm)). If this develops into a trend, manpower issues could be a serious problem as fewer system managers and programmers enter the field although roughly 10% of NIH money is anticipated to be used for bio-computing activities in the coming decade.
7. Desirable productivity tools that address the following issues:
  - Code optimization is a substantial issue. Currently available profiling tools often do not provide fine enough detail to provide significant performance improvements, even when they are possible. The profiler SHARK (Mac OSX) has demonstrated that there are substantial opportunities for optimizations that have been missed in our hand-coding efforts. Better freely available optimization/profiling tools would be very useful.
  - If MPI-based software is relied on, there are severe fault tolerance problems, since a single node crashing will kill the entire job. More fault-tolerant parallelism would be desirable.
  - Virtually all high-level programming is now done in Python with numerically intensive operations performed by embedded C++ libraries. While Python is not currently used directly

for numerically intensive work it would be quite desirable if this were possible in certain cases. The availability of a higher performance python solution is highly desirable, i.e. – a python compiler or better JIT environment.

- It is possible to write code that is relatively platform independent. Experience has shown that it takes about two days worth of effort to port a moderately sized program across platforms. However, this assumes that minimal platform-specific optimizations are performed.

### **8.12 James B. Bassingthwaight, Ph.D.**

Professor, Department of Bioengineering

University of Washington

Email: [jbb@nsr.bioeng.washington.edu](mailto:jbb@nsr.bioeng.washington.edu)

Home page: <http://depts.washington.edu/bioe/people/bassingthwaighte.shtml>

Phone: (202) 685-2012

Dr. James Bassingthwaighte is a Professor in the Department of Bioengineering at the University of Washington. He is the director of the National Simulation Resource Facility for Circulatory Transport and Exchange, which operates as a part of the Department of Bioengineering in the School of Medicine and the College of Engineering. The Resource was created with a focus on studying complex biological systems and networks involved in the transport and exchange of solutes and water in the microvasculature, within whole organs, and within the whole body. Dr. Bassingthwaighte was enthusiastic about JSim, a software environment being developed at National Simulation Resource (NSR) for scientific modeling that provides tools for development of models, for their run-time control, and for analysis of their outputs. He also discussed several issues regarding multi-scale modeling and model visualization.

#### **Summary of key points**

1. Currently, supercomputing facilities don't lend themselves to interactive use. Jobs are treated as batch processes. This does not allow the researcher to use computers as "mind expanders," interacting with the computer to design, test, run, and visualize model solutions. JSim has been used to design physiological models, reaction diffusion models,

circulatory and respiratory models, and models incorporating feedback for control of blood pressure, and large numbers of models for enzymatic reactions, channels, ionic pumps, biochemical systems and cellular excitation and contraction.

2. The JSim takes advantage of the Mathematical Modeling Language (MML) to parse equations and to determine how variables should be calculated. Utilizing a numerical library, the JSim can disentangle complex heterogeneous problems, mixing and matching algorithms as needed. JSim's numerical methods include, among others, the ODE solvers CVODE, LSODE, LSODA, Radau, and Dopri5, and the PDE solvers Tom 690 and Toms731. It also includes root solvers, delay lines, and matrix manipulators. While these solvers are among the best available, it is possible to find conditions which the solvers are unable to handle.
3. In its current form, JSim is downloaded onto a host computer and runs on a single processor. In development is a web server to allow researchers to use JSim on a client/server basis. Future versions may also allow Jsim to run in a distributed environment.
4. Multi-scale models are extremely complex. They incorporate information from the molecular and cellular levels up to organ and systems levels. Realistic models of the circulatory and respiratory systems under stress, for example exercise, require a description of the cellular events that create demands for oxygen. Having cellular level equations together with circulatory exchanges makes the system very stiff. Brute force methods are possible, but computationally demanding. It is equally challenging, however, to simplify models by using the results from the cellular level as descriptors to drive the higher level equations; changes at the higher level, for example start and stop of exercise, must be communicated back to the basic model. The development of strategies to automate the switching from the simplified submodels to the more detailed realistic submodels is critical to the designing of efficient yet realistic models that encompass several hierarchical levels.
5. An interesting challenge for researchers is to develop tools to represent complex chemical networks. Such tools might allow researchers to visualize network behavior and to map networks and their products, providing information about the state of the system as parameters are changed.

### 8.13 Joachim Frank, Ph.D.

Investigator, Howard Hughes Medical Institute

Professor, School of Public Health, Biomedical Sciences

New York State Department of Health, Wadsworth Center

Joachim@wadsworth.org

Home page: <http://www.wadsworth.org/resnres/bios/frank.htm>

Phone: (518) 474-2810

Dr. Frank is a Howard Hughes Medical School Institute Investigator. He is also a Professor for the School of Public Health and Biomedical Sciences at the New York State Department of Health, Wadsworth Center. His laboratory develops methods of 3D visualization and structural analysis with the [electron microscope](#), and applies these methods to a number of important biological structures. Electron microscopy has a unique position among structural analysis methods in that it allows the bridging of a large size range in biology, from details close to atomic resolution to cellular organelles. A 3D image is formed by using a battery of techniques, including correlation analysis, multivariate statistical analysis, classification, and reconstruction. The main focus of his work is the study of the structure and function of the ribosome. Cryo-electron microscope maps of ribosome complexes in a resolution range of 10-15 Å are used to determine the dynamic process of protein synthesis.

#### Summary of key points

1. In the past, interaction with supercomputer centers has not been good. There are too many hoops to jump through before one gains access, and too little interactive capabilities.
2. Single particle reconstruction with data from electron microscopes is computer intensive. Constructing 3D images from data obtained from the microscopes is computer intensive. The data is noisy and contains multiple views of the particle from all angles. During the reconstruction process, the most computational demanding elements include the determination the orientation of the particle with respect to a template image and the refinement cycle that selects the projection that best represents the image. A typical

reconstruction must process 10,000-100,000 noisy projections. For larger particles with larger volumes and for higher resolution images, more refinement cycles are needed.

3. The best resolution obtained to date is at 8 Angstroms. At this resolution, alpha helices are visible, but beta sheets are not. At 3 Angstroms resolution, where beta sheets would be visible and most of the structure could be traced, it is estimated that 1 million images would be required. The following capabilities need to be developed to achieve 3 Angstrom resolution:

- Data collection needs to be automated. Currently, micrographs are collected on film and then scanned. Direct digital readout with sufficiently large (4k x 4k and larger) CCD cameras, and automation of the collection of images are essential for higher productivity.
- At this resolution, there will be an explosion in the volume of computation. It will be necessary to make better use of existing computational resources, perhaps by developing distributed computing pipelines.
- Algorithms need to be developed to deal with conformational heterogeneity. An underlying assumption of single particle reconstruction is that the molecules imaged have identical structure. In order to differentiate between possible conformational states, classification algorithms, including self-organized maps and neural networks, may be needed to be folded into existing refinement techniques.

4. Some current bottlenecks in productivity include:

- Scheduling tools use a lot of overhead, so these are not used. This means that the good will of users is required when multiple users are on the system and competition for resources is high.
- Current code base is in Fortran. Investing resources to re-write the system in modern code would be desirable but is expensive and time consuming.
- There have been problems, with vendors and academic software suppliers, in dealing with different architectures. Support systems do not carry across different systems and different vendors. In addition, providing support for external institutions using their software (SPIDER) on a continuing basis is difficult. External users across multiple architectures rely on updates from a small support staff.
- Data streams and formats are not homogenous across commercial and academic software. This can present problems for collaborations, but it also prevents academic

groups from using multiple softwares. Researchers must learn to work together. One approach being tried is to divide the reconstruction procedure into interchangeable modules, with standard inputs and outputs, that are well defined and well documented.

5. Algorithmic improvements are handled by specialists. Additional improvements in the reconstruction procedure are expected to come from (i) the inclusion of classification algorithms, (ii) a refinement scheme that is based on a simultaneous solution of the reconstruction and alignment problem, and (iii) updating algorithms as platforms change to take advantage of improved performance offered by new platforms.

6. In the future, the single particle reconstruction approach will be increasingly used to look at dynamics. The ribosome must be seen as a dynamical system. In order to characterize the system over time, researchers will want to take many snapshots of system at various time points. It will be challenge to deal with the quantity of images, to link the density maps, and to interpret the time evolution of the system. This project will require much increased computational resources and different computational aspects, for example modeling and animation based on the time-evolving data.

#### **8.14 Ron Elber, Ph.D.**

Professor , Department of Computer Science

Cornell University

ron@cs.cornell.edu

Home page: <http://www.cs.cornell.edu/ron>

Phone: (607) 225-7416

Dr. Ron Elber is a Professor in the Department of Computer Science at Cornell University. He is also on the faculty of the Cornell Genomics Initiative, Computational and Statistical Genomics Focus Area. He is active in two core areas of research: bioinformatics and molecular dynamics. In bioinformatics, he is interested in protein annotation (structure and function prediction), protein evolution, protein folding potentials, and protein alignment. In protein dynamics, he develops theory, algorithms, and computer code to simulate bio-molecular dynamics, the long dynamics of biophysical processes, and protein folding. Among the substances that have been

studied in detail by Dr. Elber are the oxygen transport proteins hemoglobin and myoglobin and ion channels such as gramicidin.

### **Summary of key points**

1. Bioinformatics and molecular dynamics present different computational demands. A molecular dynamic simulation of protein folding with a medium size protein of 150 amino acids and from 1000 to 100000 particles can run for a month on a cluster of 100 off-the-shelf CPU's. On the other hand, most bioinformatics applications are much more rapid, completing within minutes or hours. While molecular dynamic simulations tend to require raw processing power, bioinformatic computation places a premium on memory and data sharing. These differences in computational needs mean that compromises must be met when purchasing new hardware.

2. The different computational needs between bioinformatics and molecular dynamic applications can also be seen at the memory usage and I/O performance levels. Bioinformatic applications depend on access to large quantities of data and often load gigabytes of data into memory at once. In addition, in a distributed environment, the data must be shared across all nodes and each node must be capable of storing at least 30 gigabytes. Molecular dynamic applications have limited I/O and memory demands, requiring on the order of 100 megabytes.

3. There is room for improvement in the following areas:

- Fault tolerance in a distributed environment. Although local, in house fixes have adequately met current needs, O/S level changes would be appropriate.
- Platform porting. With special care, code can be made portable, especially across alternative unix/linux flavors. Windows, however, presents additional challenges, especially with stability.
- Code management. Current code management tools have been found to be too restrictive in an academic setting and are convenient to use. Current applications have  $10^5$ - $10^6$  lines of code.
- Debugging tools. Productivity would improve with better debugging tools.

- Algorithmic. Some machine learning tools manipulate matrices with  $10^{16}$  elements. There is little work being done to develop algorithms to manipulate very large data sets in a distributed environment.

4. One of the striking observations in dynamics of biological molecules is the extremely large time scale they covered. Initiation by light absorption of biochemical processes is very rapid (femtoseconds), while protein folding is slow (milliseconds to minutes). Current molecular dynamic approaches are restricted to nanoseconds ( $10^{-9}$  seconds). Multi-scale modeling must maintain the detail description at the molecular level but be capable of generating a description of macro-level biology. Even if computer performance increases by a factor of two each year, this will be outpaced by the tremendous advantages that can be obtained by working on theory and algorithms, which are capable increasing performance by a factor of millions.

5. Current directions in bioinformatics will soon require that very large databases stored at multiple sites are able to be accessed, placing large demands on I/O and memory. Stability of systems, especially windows-based, will become a larger issue. Within five years, he would like to be able to access databases that are a factor of 1000 times larger than currently in use. It will then be possible to begin to answer more challenging questions about the nature of the interaction between genetic changes at the molecular level and the environment. Researchers will be able to correlate protein structure and genomic information with the different observed phenotypes. This will allow us to gain a better understanding of the interaction among species and life on earth.

### **8.15 Steinar Hauan Ph.D.**

Professor of Chemical Engineering

Carnegie Mellon University

hauan@cmu.edu

Home page: <http://www.cmu.edu/bme/faculty/hauan.html>

Phone: (412) 268-4390

Dr. Steinar Hauan is a Professor of Chemical Engineering in the Biomedical Engineering Department at the Carnegie Mellon University. Professor Hauan's research is in the area of

computer-aided process design and analysis of complex chemical systems. An important part of the work is to investigate how qualitative insights may be combined with numerical studies in order to arrive at solutions that are more readily understood. He discussed his research on agents for distributed, asynchronous process design. This research area explores how to solve large and truly hard Engineering Design problems with unknown structure. The main idea is to combine problem-specific insights and existing algorithms with techniques from artificial intelligence, information management, numerical mathematics and a distributed computing architecture. The approach is multi-threaded and relies on a collective of collaborating algorithmic agents implemented on our computer cluster.

### **Summary of key points**

1. In the future, protein separation systems and analytical chemical laboratories will be constructed in a microchip structure. The design of these Lab-on-a-Chip devices will be aided by iterative computer processes using computer aided design techniques to optimize the placement of sub-systems to create multiplexed microchips with complex topologies.
2. The placement problem can be formulated using reduced order models with a multi-objective, non-linear optimization function. Today's computers, and even tomorrow's, do not deliver enough performance to solve these optimization problems using conventional PDE's solvers for chips with a non-trivial number of subsystems (the solution space grows exponentially with the number of subsystems).
3. One approach to the placement problem is to use asynchronous agents that collaborate to arrive at a solution. This approach is multi-threaded, and requires the development of distributed algorithms without central control of agents and complex adaptive systems to monitor CPU time.
4. There are holes in existing algorithms that handle remote processes; current RPC code is not robust enough. Progress has been made in the Huan lab on a remote process interface to monitor and fire more than 300 processes per second. Current versions of MPI are not fault tolerant and do not scale well to grid processing. MPI was not designed to be fault-tolerant for asynchronous system; rather, it was implemented to enforce synchronization and would thus never get (or need) the type of redundancy and fault

tolerance necessary for large scale, asynchronous processing. Management tools are needed in a distributed environment to recover, or at least ignore, failures in communication.

5. Compiler speed and CPU types compare differently for different systems. There is no correlation between compilers and applications. The work done per cycle on the same hardware varies with compilers. Speed and performance varies across systems and compilers. It would help if someone had a benchmark library for different types of calculations. This would enable users to better evaluate what machines they should purchase and use, based on their specific applications.

6. Two important classes of algorithms that will play a large role in biomedical research are finite element code and density functional calculations. While finite elements methods are a slow way of doing optimization, they are good for modeling; they can be applies to investigate features for the Lab-on-a-Chip design. Density functional theory is being used, for example, to investigate how molecules interact with on-chip devices and to explore how blood coagulates. It will be important to lower the barrier for researchers to use these tools and provide parallelized version of existing code.

7. PDE optimization will require algorithms with improved performance. Currently, research on chip design is good with modeling at the PDE level. There is a need, however, to integrate PDE based optimization with logic based optimization and to move this integration to large, complex systems.

### **8.16 John Yate, III, Ph.D.**

Professor, Cell Biology, Department of Cell Biology

Scripps Research Institute

**[jyates@scripps.edu](mailto:jyates@scripps.edu)**

Home page: <http://www.fields.scripps.edu>

Phone: (858) 784-8863

Dr. John Yates is a Professor of Cell Biology at the Scripps Research Institute, where he is director of the Proteomics Mass Spectrometry Lab. Tandem mass spectrometry is a powerful technique for characterizing a proteome. Proteomics by tandem mass spectrometry requires

powerful informatics capabilities. First, the sequence corresponding to each peptide's tandem mass spectrum must be identified. Once those identifications have been completed, additional tools are needed to summarize and organize these identifications. Proteomics by tandem mass spectrometry requires powerful informatics capabilities.

### **Summary of Key Points**

5. Proteomic analysis using tandem mass spectrometry relies, in part, on software to automate the process of performing protein identification and peptide sequencing by utilizing mass spectrometry fragmentation patterns to search protein and nucleotide databases.
6. With improved throughput obtained by parallelizing software to run on a Beowulf cluster, one to two million spectra can now be analyzed in a week. An analysis of 100-200,000 mass spectra, which used to run for weeks to months, can now be done in hours to days, depending on the database being searched.
7. To drive down the cost of I/O operations, copies of sequence databases are stored locally. The growth in size of sequence databases will eventually stress memory capabilities.
8. The current algorithmic bottleneck occurs in the initial pass through the sequence database to identify amino acid sequences that match the measured mass of peptides under consideration. More efficient search algorithms could increase productivity by a factor of 10 to 100.
9. Space and cooling are significant cost factors for Beowulf clusters.
10. Collaboration with other research institutes could be facilitated with higher bandwidth internet communications. Experience has shown that it is often faster to run an analysis on a mass spectrometer dataset on slower hardware than it is to ftp that same dataset to another site.

## Conclusion

Computing requirements are dramatically increasing in all areas of biological research. Current and future needs will focus on integration of diverse sets of data, originating from a variety of experimental techniques which are capable of producing data at the levels of entire cells, organs, organisms and populations. Our study found a critical need for theoretical, algorithmic and software advances in storing, retrieving, networking, processing, analyzing, navigating and visualizing biological information. Sophisticated machine learning approaches are needed in order to deal with huge amounts of data. Machine learning methods (neural networks, hidden Markov models, etc.) are well suited for domains characterized by large quantities of data, noisy patterns and the absence of general theories. These methods are computationally intensive, clearly require high-end computing capabilities, and would benefit from further improvements in computational performance.

## References

- 
- <sup>1</sup> Report of the Bioinformatics Working Group of the National Advisory Research Resources Council, June 2000.
- <sup>2</sup> Baldi, Pierre and Brucak, Soren. *Bioinformatics The Machine Learning Approach* MIT Press 2001. p. xi
- <sup>3</sup> Virtual lung models every breath you take.-and its impact. 27 September 2001.  
<<http://www.pnl.gov/news/2001/01-33.htm>>
- <sup>4</sup> R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic models of proteins and nucleic acids*. Pp. 51-68. Cambridge University Press. 1998.
- <sup>5</sup> <http://www.ncbi.nlm.nih.gov/>
- <sup>6</sup> <http://cbcg.llbl.gov/ssi-csb/Chapter2.html>
- <sup>7</sup> <http://www.paracel.com/products/index.html>
- <sup>8</sup> S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J. Mol. Biol*, 215:403-410, 1990
- <sup>9</sup> [http://www-cse.stanford.edu/classes/sophomore-college/projects-00/computers-and-the-hgp/smith\\_waterman.html](http://www-cse.stanford.edu/classes/sophomore-college/projects-00/computers-and-the-hgp/smith_waterman.html)
- <sup>10</sup> R. Hughey and A. Krogh. Hidden Markov models for sequence for analysis:extension and analysis of the basic method. *Computer Applications in the Biosciences*, 12:95-107. 1996.
- <sup>11</sup> Rabiner, L.R. "A Tutorial on Hidden Markov Models and Selected Application in Speech Recognition", *Proceedings of the IEEE*, Vol. 77, pp. 257-286, February 1989.
- <sup>12</sup> Kevin Karplus, Christian Barrett, Richard Hughey, "[Hidden Markov Models for Detecting Remote Protein Homologies](http://www.cse.ucsc.edu/research/compbio)", *Bioinformatics* 14(10):846-856, 1998. WWW server available from <http://www.cse.ucsc.edu/research/compbio>.
- <sup>13</sup> Baldi, Pierre and Brunak, Soren. *Bioinformatics The Machine Learning Approach*. Pg. 297. MIT Press 2001.
- <sup>14</sup> Desper, R., Gascuel, O. (2002). Fast and Accurate Phylogeny Reconstruction Algorithms Based on the Minimum-Evolution Principle, *Proceedings of the 2<sup>nd</sup> Workshop on Algorithms in Bioinformatics (WABI)*, Roma, Lecture Notes in Computer Science.
- <sup>15</sup> Felsenstein, J. (1989). PHYLIP - Phylogeny inference package (version 3.2). *Cladistics*. 5: 164-6.
- <sup>16</sup> Lawrence, C.E., Altschul, S.F., Bogouski, M.S., Liu, J.S., Neuwald, A.F., and Wooten, J.C., "Detecting Subtle Sequence Signals: A Gibbs Sampling Strategy for Multiple Alignment," *Science*, 262, 208-214 (1993).
- <sup>17</sup> Aha, D.W., and Bankert, R.L. (1996). A comparative evaluation of sequential feature selection algorithms. In D. Fisher and J.-H. Lenx (Eds.), *Artificial Intelligence and statistics V*. New York: Springer-Verlag.
- <sup>18</sup> Yang, J., and Honavar, V. (1997). Feature subset selection using a genetic algorithm. *Proceedings of the Genetic Programming Conference*, pages 380-385. Stanford, CA.
- <sup>19</sup> Glover, F., Laguna, M., and Marti, R. Scatter search. *Theory and Applications of Evolutionary Computation: Recent Trends*. A. Ghosh and S. Tsutsui (Eds.), Springer-Verlag.

- 
- <sup>20</sup> Shah DC, Kusiak A., (2004) Data mining and genetic algorithm bases gene/SNP selection. *Artif. Intell. Med.* Jul; 31(3) 183-96.
- <sup>21</sup> Miyaki K, Omae K, Murata M, Tanahashi N, Saito I, Watanabe K., High throughput multiple combination extraction from large scale polymorphism data by exact tree method. *J. Hum Genet* (2004) Aug 11
- <sup>22</sup> Alain Vignal, Denis Milan, Magali SanCristobal and André Eggen A review on SNP and other types of molecular markers and their use in animal genetics *Genet. Sel. Evol.* 34 (2002) 275-305 DOI: 10.1051/gse:2002009
- <sup>23</sup> MacKay, D. J. C. (1992). "Bayesian Methods for Adaptive Methods." Ph.D. thesis, California Institute of Technology.
- <sup>24</sup> Muller, P. and Rios Insua, D. (1998). "Feedforward Neural Networks for Nonparametric Regression." In *Practical Nonparametric and Semiparametric Bayesian Statistics*, eds. D. Dey, P. Müller, and D. Sinha. New York: Springer-Verlag.
- <sup>25</sup> Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press.
- <sup>26</sup> Akaike, H. (1974). "A New Look at Statistical Model Identification." *IEEE Transactions on Automatic Control*, AU-19, 716-722.
- <sup>27</sup> Schwarz, G. (1978). "Estimating the Dimension of a Model." *The Annals of Statistics*, 6, 2, 461-464.
- <sup>28</sup> Murata, N, Yoshizawa, S., and Amari, S. (1994). "Network Information Criterion—Determining the Number of Hidden Units for an Artificial Neural Network Model." *IEEE Transactions on Neural Networks*, 5, 6, 865-871.
- <sup>29</sup> MacKay, D. J. C. "Bayesian Non-Linear Modeling for the Energy Prediction Competition." (1994) *ASHRAE Transactions*, 100, pt. 2, 1053-1062.
- <sup>30</sup> Lee SM, Abbot PA. Bayesian networks for knowledge discovery in large datasets: basics for nurse researchers. *J Biomed Inform.* 2003 Aug-Oct;36(4-5):389-99.
- <sup>31</sup> Bate A, Lindquist M, Edwards IR, Olsson S, Orre R, Lanser A, De Freitas RM. A Bayesian neural network method of adverse drug reaction signal generation. *Eur. J. Clinical Pharmacol.* (1998) Jun; 54(4):315-321
- <sup>32</sup> "New Challenges in Computational Biochemistry", B .Honig, Department of Biochemistry and Molecular Biophysics, Columbia University ,630 West 168 St. ,New York, NY 10032
- <sup>33</sup> Rapaport, D.C. *The Art of Molecular Dynamics Simulation*. Cambridge University Press. 1995.
- <sup>34</sup> Almlof J, K Faegri and K Korsell 1982. Principles for a Direct SCF Approach to LCAO-MO *Ab initio Calculations*. *Journal of Computational Chemistry* 3: 386-399.
- <sup>35</sup> Leach, Andrew R. *Molecular Modeling :Principles and Applications*. Pearson Education Unlimited: Essex, England, 2001.
- <sup>36</sup> Clark D. E and D. R. Westhead 1996. Evolutionary Algorithms in Computer-Aided Molecular Design. *Journal of Computer-Aided Molecular Design*. 10:337-358.
- <sup>37</sup> Baldi, Pierre and Soren Brunak. 2001. *Bioinformatics: the machine learning approach*. Pg. 91. 2<sup>nd</sup> ed. The MIT Press.
- <sup>38</sup> <http://www.pnl.gov/news/2001/01-33.htm>

- 
- <sup>39</sup> McQueen, D.M., and C.S. Peskin. 2000. A three-dimensional computer model of the human heart for studying cardiac fluid dynamics. *Computer Graphics* 34:56–60.
- <sup>40</sup> McQueen, D.M., and C.S. Peskin. 1997. Shared-memory parallel vector implementation of the immersed boundary method for the computation of blood flow in the beating mammalian heart. *Journal of Supercomputing* 11(3):213–236.
- <sup>41</sup> <http://www.npaci.edu/SAC/Collaborations/Peskin/reports/initial.profile.html>
- <sup>42</sup> <http://www.npaci.edu/envision/v16.4/adaptivecomp.html>.
- <sup>43</sup> Yelick, K., L. Semenzato, G. Pike, C. Miyamoto, B. Liblit, A. Krishnamurthy, P. Hilfinger, S. Graham, D. Gay, P. Colella, and A. Aiken. 1998. Titanium: A High-Performance Java Dialect. *Concurrency: Practice and Experience*, September–November 1998:825–36.
- <sup>44</sup> Hiroaki Kitano, *Foundations of Systems Biology*, p. 1.
- <sup>45</sup> H. H. McAdams and A. Arkin. *Simulation of Prokaryotic Genetic Circuits*. *An. Rev. Biophys. Biomol. Struct.*, 27:199–224, 1998
- <sup>46</sup> M. D. Levin, T. S. Shimizu, and D. Bray, Binding and Diffusion of CheR Molecules Within a Cluster of Membrane Receptors *Biophys. J.*, April 1, 2002; 82(4): 1809 - 1817.
- <sup>47</sup> BIOSPICE. (2001). The BioSPICE Development Project [On-line]. Available: [www.biospice.org/](http://www.biospice.org/).
- <sup>48</sup> Goryanin I, Hodgman TC, Selkov E. (1999) Mathematical simulation and analysis of cellular metabolism and regulation. *Bioinformatics* 1999, 749-58.
- <sup>49</sup> Mendes, Pedro. 1997. Biochemistry by numbers: Simulation of biochemical pathways with Gepasi 3 . *Trends in Biochemical Sciences* 22: 361-363.
- <sup>50</sup> Metabolic control and its analysis. Extensions to the theory and matrix method. *Eur J Biochem.* 1987 May 15;165(1):215-21.
- <sup>51</sup> Bray, D., Levin, M.D., Morton-Firth, C.J.: *Receptor clustering as a cellular mechanism to control sensitivity*. *Nature* 393 (1998) 85-88.
- <sup>52</sup> Loew, Leslie M., and Jim C. Schaff. 2001. The Virtual Cell: A software environment for computational cell biology . *Trends in Biotechnology* 19: 401-406.
- <sup>53</sup> A. Finney and M. Hucka, Systems biology markup language: Level 2 and beyond *Biochem. Soc. Trans. (2003)* 31, (1472–1473).
- <sup>54</sup> Grasshoff, K., M. Ehrhardt, and K. Kremling, A Benchmark for Methods in Reverse Engineering and Model Discrimination: Problem, *Genome Res.* 2004; 14: 1773-1785.
- <sup>55</sup> Van Ginkel, J.H., Gorissen, A. and Polci, D. 2000. Elevated atmospheric carbon dioxide concentration: effects of increased carbon input in a *Lolium perenne* soil on microorganisms and decomposition. *Soil Biology & Biochemistry* 32: 449-456.
- <sup>56</sup> Mangold, H. K. (1964) *J. Am. Oil. Chem. Soc.* 47, 762–777
- <sup>57</sup> Shimizu, Thomas Simon and Dennis Bray, *Computational Cell Biology – The Stochastic Approach*, p. 213.

- 
- <sup>58</sup> Gillespie, D. T. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.* 22, 403-434.
- <sup>59</sup> Ehldé M, Zacchi G. MIST: a user-friendly metabolic simulator. *Comput Appl Biosci.* 1995 Apr;11(2):201-7.
- <sup>60</sup> Sauro, H. M., (1993) SCAMP: a general-purpose simulator and metabolic control analysis program. *Comput Appl Biosci.* 9 (4), 441-450.
- <sup>61</sup> Uhley, J., Wilson, M., Bhalla, U., & Bower, J. (1990). A UNIX, X-windows-based neural network simulation system. In *USENIX '90 Conference Proceedings*
- <sup>62</sup> Hines, M.L. and Carnevale, N.T., The NEURON simulation environment, *Neural Computation* 9, 1179-1209 (1997).
- <sup>63</sup> Leach, Andrew R. Molecular Modelling :Principles and Applications. Pearson Education Unlimited: Essex, England, 2001. pg. 642.
- <sup>64</sup> J.R. Ullman. *An algorithm for subgraph isomorphism.* Journal of the ACM, 23(1):31--42, 1976.
- <sup>65</sup> Thornber, C.W. 1979. Isosterism and Molecular Modification in Drug Design. *Chemical Society Reviews* 8:563-580.
- <sup>66</sup> Patani, G. and LaVoie, E.J. (1996) "Bioisosterism: A Rationale Approach in Drug Design", *Chemical Reviews.* 96: 3147-3176.
- <sup>67</sup> Dammkoehler, R.A. Karasek, S.F., Shands, E.F.B., and Marshall, G.R. 1989. Constrained Search of Conformational Hyperspace. *Journal of Chemical Information and Computer Science* 32:244-255.
- <sup>68</sup> Sheridan, R. P, Milakanton, R., Dixon, J.S and Venkataraghavan. 1986. The Ensemble Approach to Distance Geometry: Application to Nicotinic Pharmacophore. *Journal of Medicinal Chemistry* 29:899-906.
- <sup>69</sup> Bron, C. and Kerbosch, J. 1973. Finding all Cliques of an Undirected Graph. *Communications of the ACM* 16:575-577.
- <sup>70</sup> Barnum, D., Greene, J., Smellie, A. and Sprague, P. 1996. Identification of Common Functional Configuration among Molecules. *Journal of Chemical Information and Computer Science* 36:563-571.
- <sup>71</sup> Goodsell, D.S. and Loson, A.J. 1990. Automated Docking of Substrates to Proteins by Simulated Annealing. *Proteins: Structure, Function and Genetics* 8:195-202.
- <sup>72</sup> Jones, G., Willett, P., Glen, R.C., Leach R. and Taylor R., 1997. Development and Validation of a Genetic Algorithm for Flexible Docking. *Journal of Molecular Biology* 267:727-748.
- <sup>73</sup> Rarey M, Krammer B., Lengauer and Klebe, G. 1996. A Fast Flexible Docking Methods Using an Incremental Construction Algorithm,. *Journal of Molecular Biology* 261: 470-489.
- <sup>74</sup> Meng E.C, Shoichet, B.K. and Kuntz, I.D. 1992. Automated Docking with Grid-Based Energy Evaluation. *Journal of Computational Chemistry.* 13:505-524.
- <sup>75</sup> Chaifson, P.S., Corkery, J.J., Murcko, M.A. and Walter, W.P. 1999. Consensus Scoring. A Method for Obtaining Improved Hit Rates from Docking Databases of Three-Dimensional Structures into Proteins. *Journal of Medicinal Chemistry* 42:5100-5109.
- <sup>76</sup> Kuntz et al., *Journal of Molecular Biology.* Vol. 161, pp. 269.

---

<sup>77</sup> Kuntz, *et al.*, *J. Mol. Biol.*, **161**, 269, 1982.

<sup>78</sup> Jones et al., *Journal of Molecular Biology* 267:727-748, 1997.

<sup>79</sup> Lewis, R.M. and Leach A. R, 1994. Current Methods for Site-Directed Structure Generation. *Journal of Computer-Aided Molecular Design* \*:467-475.

<sup>80</sup> S. Parker, "A component-based architecture for parallel multi-physics PDE simulation," presented at Presented at International Conference on Computational Science (ICCS2002) Workshop on PDE Software, 2002.

<sup>81</sup> J. E. Guilkey and J. A. Weiss, "Implicit time integration for the Material Point Method: Quantitative and algorithmic comparisons with the Finite Element Method." *International Journal for Numerical Methods in Engineering*, Vol. 57(11), July Issue, 2003.