Pacific Symposium on Biocomputing 6:360-371 (2001)

USING METACOMPUTING TOOLS TO FACILITATE LARGE-SCALE ANALYSES OF BIOLOGICAL DATABASES

Allison Waugh[1], Glenn A. Williams[1], Liping Wei[2], and Russ B. Altman[1]
[1]Stanford Medical Informatics
251 Campus Drive, MSOB X-215, Stanford, CA 94305-5479
{waugh, gaw, altman} @ smi.stanford.edu
[2]Exelixis, Inc.
260 Littlefield Ave, South San Francisco, CA 94080
lwei@exelixis.com

ABSTRACT

Given the high rate at which biological data are being collected and made public, it is essential that computational tools be developed that are capable of efficiently accessing and analyzing these data. High-performance distributed computing resources can play a key role in enabling large-scale analyses of biological databases. We use a distributed computing environment, LEGION, to enable large-scale computations on the Protein Data Bank (PDB). In particular, we employ the FEATURE program to scan all protein structures in the PDB in search for unrecognized potential cation binding sites. We evaluate the efficiency of LEGION's parallel execution capabilities and analyze the initial biological implications that result from having a site annotation scan of the entire PDB. We discuss four interesting proteins with unannotated, high-scoring candidate cation binding sites.

INTRODUCTION

In the last three years the Protein Data Bank (PDB) has increased from 5,811 released atomic coordinate entries to 12,110[1]. The growth of the PDB is typical of many other biological databases. As these databases increase in size, analyzing their contents in a systematic manner becomes important. Two common analyses are (1) linear scans that examine each entry (such as looking through all DNA sequences for the presence of a new motif), and (2) "all-versus-all" comparisons in which each element is compared with all other elements (such as clustering of structures, as reported in Shindyalov and Bourne[2]). To make some analyses tractable, investigators have created smaller, "nonredundant" subsets of these databases[3]. However, the growth of these databases coupled with the interest in looking at all entries makes the use of large parallel computers or distributed computational systems inevitable.

Distributed and parallel computing has been applied to the computationally intensive domain of molecular dynamics simulations[4,5,6,7]. Bywater et al.[8] describe a high performance distributed compute server for automatic functional annotation of protein sequences, three-dimensional protein structure prediction tools, and protein comparisons. Yap et al.[9] demonstrate that parallel computational methods can

significantly reduce computation time in sequence analysis. Shindyalov and Bourne[2] compared more than 8,000 proteins in the PDB against each other. This effort consumed more than 24,000 processor hours on the Cray T3E at the San Diego Supercomputer Center (SDSC) and would have required almost three years of CPU time on a single workstation[2]. However, high-performance computing systems generally call for special expertise, such as the skill required to parallelize an algorithm, and are not yet used regularly by the biocomputing community.

FEATURE is a site-characterization and recognition system that identifies functional or structural sites of interest in a query protein[10]. A site is defined as a microenvironment, distinguished by some structural or functional role, within a biomolecular structure[11]. FEATURE measures spatial distributions of chemical and physical properties (including atomic, chemical group, residue, and secondary structure features) to create statistical models of microenvironments. It compares regions of a query protein with known sites and control nonsites (regions that do not have the particular structural or functional role of the site) and assigns a score indicating the likelihood that the local region is a site. The method produces a list of potential site locations and their scores. FEATURE has been shown to recognize ion binding sites, small molecule ligand binding sites, and enzyme active sites[11,12,10].

FEATURE is typical of many data-driven algorithms, requiring both large data storage and efficient data analysis. On standard workstations, FEATURE has required 12 hours on a single processor to evaluate 580 non-redundant PDB entries[13]. We require 20 to 100 times speed-up to allow routine scans of the entire PDB.

Recently, computational grid systems have emerged to provide distributed, pervasive, dependable, and consistent access to high-performance computational resources. These systems contain software layers that transform collections of independent resources into a single, coherent, virtual machine accessible from one workstation. Metasystems currently being developed include Globus[14,a], Globe[15], Condor[16,b], Metacomputer Online (MOL)[17,c], and Polder[d].

The experiments described here were performed on LEGION, one of the metacomputing environments supported by the National Partnership for Advanced Computational Infrastructure (NPACI)[e] at SDSC. LEGION[f] is an object-based metasystem comprised of geographically distributed, heterogeneous collections of workstations and supercomputers[18,19]. Lying atop a user's operating system, LEGION

---

[a] http://www.globus.org/
[b] http://www.cs.wisc.edu/condor
[c] http://www.uni- paderborn.de/pc2/projects/mol
[d] http://www.science.uva.nl/projects/polder
[e] http://www.npaci.edu/
[f] http://www.legion.virginia.edu/

provides features and services necessary to schedule and distribute the user's task on available and suitable hosts, allowing the user to take advantage of large, complex resource pools. When LEGION schedules tasks over multiple remote machines, it automatically transfers the appropriate binaries to each host, eliminating the need to move and install binaries manually on multiple platforms. LEGION also provides a single, persistent namespace that spans all machines in a LEGION network. When a computation is divided across multiple machines that do not share a common file system, input and output files are still available to all parts of a computation.

The goal of our work is to prototype a bioinformatics infrastructure for performing routine, large-scale analyses. Specifically, the objectives of this paper are to: (1) take advantage of available high-performance distributed computing resources to make time-efficient scans of all known protein structures in search of potential sites of biological interest, (2) evaluate the performance of scans made using distributed computing resources and compare to that of sequential scans made using a single-processor workstation, and (3) analyze the results of the parallel scans and comment on the biological implications that were elucidated.

METHODS

We used LEGION to apply FEATURE to all protein structure entries in the May 2000 release of the PDB. In these scans, we searched for potential calcium binding sites, a site-type for which FEATURE has been shown to be accurate in recognizing. It has a sensitivity of 90% and specificity near 100% in cross-validation analyses on known calcium binding sites and control nonsites[10,20]. FEATURE accepts as input a PDB file and a DSSP file. DSSP is used to extract secondary structure and surface assignments for all protein entries[21]. We examined 11,956 PDB entries and 11,043 DSSP entries. The complete PDB and DSSP databases were stored on an Intel 686 machine (Pathfinder) at the University of Virginia.

We performed three types of experiments using FEATURE to search for calcium binding sites: (1) as a baseline, we sequentially scanned a PDB subset on a single processor, (2) we ran a comprehensive scan of all PDB proteins using the LEGION system, and (3) we performed a set of runs on LEGION using a constant PDB subset while varying the number of concurrent processes, the results of which were used to evaluate how well the scans scaled over the distributed computing environment.

In all the experiments, the input parameters to FEATURE and the statistical model of the calcium binding site remained constant. We used a model of calcium binding that was generated from an analysis of the spherical regions of radius 7.5 Å around 59 known calcium binding sites and 140 control nonsites. The sampling size of the grid used in scanning query proteins was 1.25 Å. We have observed that false positives detected by FEATURE are often magnesium or other divalent cation

binding sites that are difficult to distinguish from true calcium binding sites. For our complete scan of the PDB, we applied a high score threshold to minimize false positives and enrich for sites that are very likely to bind either calcium or more generally some other cation. We employed the PDBsum database[22] to create a list of cations that might cause high scores in our model. This list was used in evaluating FEATURE's ability to detect cation binding sites, as seen in Table 2.

In the first experiment, we used FEATURE to scan sequentially 726 proteins arbitrarily selected from the PDB. The runs were made on one processor of a Sun E450 machine with a 300 MHz Ultra-Sparc CPU, with the necessary PDB and DSSP files stored on the local disk.

The remaining sets of runs employed LEGION. LEGION version 1.6.5 for a SunOS required approximately 200 MB of disk space to store LEGION binaries and setup scripts. The FEATURE code was compiled on LEGION for Intel Linux, DEC Alpha Linux, and Sun Solaris environments and the resulting binaries were registered into "LEGION space." To run the FEATURE code on LEGION, we called the LEGION command, *legion_run_multi*, which spawns multiple instances of the code to available and suitable machines. The *legion_run_multi* command allows the user to control the maximum number of processes, *np*, running concurrently in LEGION. We tested values from 20 to 80. A specification file passed to the *legion_run_multi* command provides instructions on where to retrieve input files, which input files are constant for all runs, which input files vary, and where to deposit the resulting output files. The *legion_run_multi* command employs pattern matching to cycle through the variable input files that are individually fed to different runs of FEATURE and to produce corresponding output files. We restricted the FEATURE class on LEGION to "interactive" mode, ensuring that jobs would not be placed on a queue.

The second experiment was a comprehensive scan of all proteins in the PDB. 10,996 structures had entries in both the PDB and DSSP. The scan was made on Pathfinder using LEGION version 1.6.5. The maximum number of processes running simultaneously within the LEGION system was set to 50.

The third experiment was made using LEGION version 1.6.5 launched from a Sun E450 machine. The purpose of this experiment was to obtain timings of *legion_run_multi* scans using a range of different *np* values. From these times we evaluated how LEGION scales with respect to the maximum number of processes running simultaneously. We arbitrarily selected 4,997 of all the structures in the PDB for FEATURE to evaluate. We recorded the time it took to scan this set of structures using *np* values of 20, 40, 60, and 80.

Table 1: Time results from a FEATURE scan of 720 proteins, run sequentially on a single processor, and from the comprehensive FEATURE scan of the PDB on LEGION. "Estimated CPU time" refers to the time that would be required to scan sequentially the PDB on a single processor. The number of atoms is used as a basis for extrapolation because FEATURE scales linearly with the number of atoms.

Sequential Scan of PDB Subset		**Legion Scan of Complete PDB**	
Number of Proteins	720	Number of Proteins	10,911
Number of Atoms	2,613,560	Number of Atoms	37,300,301
Total CPU Time (Hrs)	12.4	**Estimated CPU Time** (Hrs)	177
CPU Seconds/Atom	0.017	**Actual Clock Time** (Hrs)	10.6

RESULTS

Results of the sequential scan are shown in Table 1. In this scan, FEATURE reported six run-time failures due to non-standard PDB file formats and compile-time memory restrictions while evaluating the 726 proteins. Table 1 also displays time results of the second experiment. Because 17 FEATURE runs failed due to LEGION file transfer errors, we had to launch the *legion_run_multi* command a second time to complete the PDB scan. Of the 10,996 proteins, FEATURE reported run-time assertion failures, illegal instructions, or segmentation faults independently of LEGION while evaluating 85 proteins. The 10,911 structures that were successfully examined consisted of 37,300,301 atoms in total. Table 1 shows an estimate of the CPU time required to sequentially scan the entire PDB and the actual elapsed clock time of the comprehensive scan. Since the FEATURE code scales linearly with the number of atoms, we obtained this estimated time from the total number of atoms successfully scanned in the first two experiments. Results of the third experiment, the scaling runs, are shown in Figure 1, which plots clock time against the maximum number of FEATURE scan processes being run simultaneously. Combining the two LEGION experiments, a total of 199 CPUs on 124 LEGION machines performed at least one FEATURE run.

Results from the analysis of the FEATURE scans on the entire PDB are provided in Table 2, which displays the number of proteins predicted to bind a cation with various score cutoffs. Scores represent the strength of prediction of potential cation binding sites. Figures 2 through 5 show cation binding predictions made by FEATURE for four proteins, discussed below. None of these proteins' PDB entries contain HETATM annotations for cations.

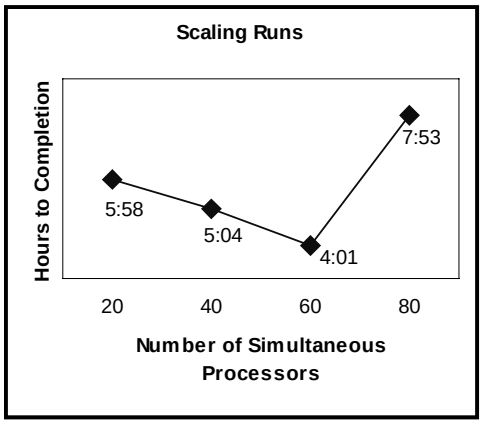

Figure 1: Time results of four FEATURE scans on 4,997 PDB entries, each scan run in parallel on LEGION. The PDB entries remained constant across all four scans. Plot shows the elapsed clock run time versus number of FEATURE instances running concurrently. LEGION scales well up to 60 simultaneous processors but then suffers a performance drop as discussed in the text.

Table 2: FEATURE results of the comprehensive PDB scan run on LEGION. Table rows correspond to different FEATURE cutoff scores for determining binding sites; higher scores correspond to stronger predictions. The second column reports the total number of cation binding proteins predicted from the scan. The third column gives the number of predicted cation binding proteins that have PDB entries containing a HETATM annotation of calcium. The fourth column shows the number of proteins with no predicted site scoring above the corresponding cutoff but which have PDB entries containing a HETATM annotation of calcium. The fifth through seventh columns report the number of predicted cation binding proteins that have PDB entries with no HETATM annotations of CA; CA or MG; any divalent cation, respectively.

Score cutoff	Total predicted	Predicted with CA annotated	No prediction With CA Annotated	Predicted with no CA annotated	Predicted with no CA or MG annotated	Predicted with no divalent cation* Annotated
20	5173	980	65	4193	3802	2951
25	3303	901	144	2402	2068	1469
30	2029	795	250	1234	982	598
35	1260	660	385	600	428	190
40	789	476	569	313	203	55

*and no HO, FE, GD, TB annotated

DISCUSSION

The major steps to preparing for a LEGION run involve (1) downloading, unpacking, and installing the LEGION binaries and setup scripts, (2) writing a LEGION makefile to compile and register the code for various architectures, and (3) creating a LEGION specification file to instruct where to receive input and deposit output. Our makefile and specification file along with the LEGION commands leading to a parallel run are available at http://www.smi.stanford.edu/projects/helix/pubs/psb01-legion/. Upon completion of these steps and compilation, the user can login to the LEGION network and begin parallel computations without having to modify any code. As expected in a network comprised of multiple platforms, the user's code must be able to compile on various architectures to take full advantage of LEGION's power. LEGION version 1.6.5 remained active throughout our experiments.

The performance of LEGION is promising. The comprehensive scan of the PDB on LEGION successfully evaluated 10,911 structures with 37.3 million atoms in less than 11 hours of clock time. A sequential scan of those proteins on a Sun 300 MHz processor machine would require approximately 177 CPU hours. Without having to parallelize the FEATURE algorithm, a significant decrease in computation time was achieved. However, the extent of parallelism of a LEGION run-multi is limited. Figure 1 demonstrates the performance drop when the maximum number of simultaneous runs is set to 80. The optimal maximum number is restricted by the size of the operating system's process table of the client machine (which maintains information on each LEGION process spawned) and the amount of memory available to support the spawned instances. Thus, although the LEGION network may contain hundreds of nodes, the user cannot perform hundreds of concurrent runs. The LEGION team is addressing this limitation.

The *legion_run_multi* command provides minimal support for fault-tolerance. As each spawned instance finishes, LEGION prints information to a client-specified output device as to the success of that run. If one or more instances fail, the user must wait for all remaining instances to be spawned and completed and then re-launch the *legion_run_multi* command. LEGION has implemented a simple method to then determine which instances failed and re-spawn those instances. It would be preferable that the LEGION network understand at run-time when a failure occurs and rerun the failed instance while others are finishing.

Maintaining a local copy of the query database, as we did in the experiments, is not a realistic solution to obtaining database entries. Local storage requires both a large amount of disk space and methods to update the local copy as new entries are added to the database. We are investigating using SDSC's Storage Resource Broker (SRB)[23,g] to develop a more robust strategy for data-transfer.

---

[g] http://www.npaci.edu/DICE/SRB

Figure 2: FEATURE's predicted locations of cation binding sites (indicated by balls) with scores above 15 in calpain domain dVI (one chain shown). These predictions map to the three metal binding sites in the EF2, EF3 and EF4 regions of dVI crystallized in high Ca^{2+}.

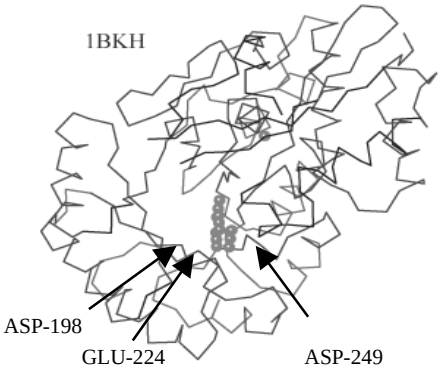

Figure 3: FEATURE's predicted locations of cation binding sites (indicated by balls) of the muconate lactonizing enzyme (one chain shown). The three residues highlighted are the metal binding ligands as reported in the Mn^{2+}-bound structure.

Table 2 presents the number of predicted cation binding proteins resulting from the FEATURE scan of the entire PDB, using different cutoff scores to indicate a potential binding site. FEATURE predicted at least one high-scoring site in 55 PDB entries that do not contain any annotations of the cations in our list generated from PDBsum. Because we used a high score-cutoff and removed all proteins with cation annotations, these entries represent possible novel discoveries of cation binding sites. Of these 55 structures, 1aj5, 1bkh, 1cjd, and 1djc are discussed below.

Figure 2 shows the locations of predicted metal binding sites in the C-terminal domain (apo-dVI) structure of the small subunit of calpain that was crystallized without Ca^{2+} (PDB entry 1aj5)[24]. With a score cutoff of 15.0, we predict three metal binding sites. These three site locations map directly to the three metal binding sites in the EF2, EF3, and EF4 regions of the dVI structure crystallized in the presence of high Ca^{2+} (PDB entry 1dvi)[24]. Blanchard et al.[24] state that the calcium binding site in EF4 failed to be identified in the crystallization of dVI when a low concentration of Ca^{2+} was used. Only in the presence of high Ca^{2+} was it seen. FEATURE was able to detect this site. There has been some controversy about whether the EF5 region binds calcium. Contrary to predictions made based on mutagenesis experiments, the 1dvi structure reveals that the EF5 region does not bind Ca^{2+} even in high concentration. FEATURE does not predict a site in the EF5 region at a cutoff of 15. Finally, FEATURE failed to recognize the calcium binding

Figure 4: FEATURE's novel predicted locations of cation binding sites (indicated by balls) in P3, the major coat protein of PRD1. The three symmetric sites sit atop the "pore", through which it may be possible for ions to pass (Benson et al., 1990a).

site in the EF1 region. The largest structural changes of dVI upon binding of calcium are associated with this EF1 region, yielding a RMS deviation of 4.34 Å for the 18 C_α atom pairs[24]. It is likely that the conformation of dVI is not suitable for calcium binding, but that an induced fit mechanism acts to bring the necessary atoms into appropriate position. Thus, FEATURE's failure to recognize this site is not surprising.

Figure 3 displays the location of a potential cation binding site in the structure of muconate lactonizing enzyme (MLE) with no bound metal ion in the crystal structure (PDB entry 1bkh)[25]. The structure of this MLE is highly similar to the determined Mn^{2+}-bound structure (PDB entry 1muc)[26]. MLE is part of an enzyme class that necessitates the binding of a divalent metal ion for catalysis. The authors attribute the lack of the metal ion in the non-metal-bound MLE structure to the crystallization drop containing 0.1 mM $MnCl_2$, whereas 2 mM $MnCl_2$ was present in the crystallization mix that produced MLE crystals bound to the metal-ion[26]. The three direct protein ligands to the metal ion in MLE are D198, E224, and D249. As illustrated in Figure 3, these ligands surround the location of FEATURE's predicted site.

Figure 4 shows locations of potential divalent metal binding sites in P3 (PDB entry 1cjd)[27], the major capsid protein of the bacteriophage PRD1. P3 is a trimer that forms a pore-like region at its center. Our three predicted sites sit symmetrically around the top of this pore. Although in a space-filling model the pore seems to completely seal the top with three valine residues (Val-134), it is possible that the pore could allow the passage of ions[27]. Furthermore, the tertiary

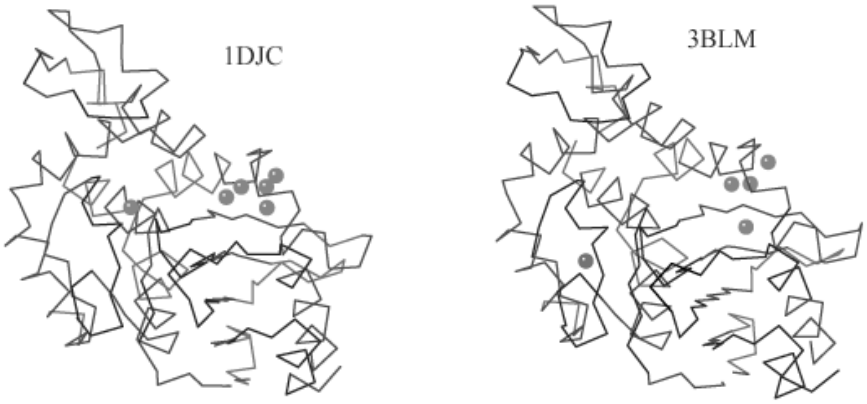

Figure 5: FEATURE's predicted locations of cation binding sites (indicated by balls) in the S70A mutant of β-lactamase and its native enzyme. Neither PDB entries report any divalent cation binding ligands. FEATURE is predicting a novel cation binding site located on the edge of the active-site depression.

structure of P3 is reminiscent of hexon (PDB entry 1dhx), the major coat protein of the mammalian adenovirus, with the additional residues of hexon mainly occurring in the loops[28]. While lacking any apparent sequence similarity, the major coat proteins possess similar functions as well as similar architecture[27]. FEATURE also predicts a cation binding site with scores above 30 in hexon. Further study is required to validate our predictions.

Figure 5 shows locations of potential cation binding sites in the S70A mutant of the β-lactamase (PDB entry 1djc)[29] and its native structure (PDB entry 3blm)[30]. The S70A mutant was manufactured by eliminating the nucleophilic group (Ser70) that attacks the beta-lactam carbonyl carbon. Other than the mutation site, the structure is identical to that of the native enzyme. Neither PDB entries report any metal binding ligands and the predicted sites in the two structures fall into the same region. In these beta-lactamases, there is a Ser70 located at the bottom of a rather shallow depression that constitutes the active site[29]. The beta-lactam does not contain a positively charged group, and so FEATURE is predicting a novel cation binding site located on the edge of the depression. This prediction requires further study to determine its validity.

Our results demonstrate that metacomputing systems, such as LEGION, enable large-scale analyses on biological databases. We have shown that the high-parallelism of LEGION allows us to run FEATURE systematically across the PDB, locating potentially interesting metal binding sites in less than half of a day. One clear use of this capability is in the emerging efforts in structural genomics. As large numbers of protein structures are produced at an ever-increasing rate, the need

to automatically annotate these structures with sites of structural and functional significance becomes acute. Our results suggest that programs such as FEATURE can be used in the context of metacomputing environments to provide rapid annotation of the new structures that emerge from these efforts.

ACKNOWLEDGMENTS

This work was supported by NSF-10152756, NSF-DBI-9600367, NIH LM-05652, LM-06422. We would like to thank the members of the LEGION team for their contributions to this work, particularly Andrew Grimshaw, Anand Natrajan, Ellen Stackpole, Norman Beekwilder, and Mark Morgan. We thank Ricky Connell for his efforts in installing and maintaining LEGION locally and Michelle Carrillo for her assistance in the biological analysis.

REFERENCES

1. Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. 2000. The Protein Data Bank. Nucleic Acids Research 28, 235-242.
2. Shindyalov, I. V., and Bourne, P. E. 1998. Protein structure alignment by incremental combinatorial extension of the optimum path. Protein Eng. 11, 739-747.
3. Hobohm,U. and Sander,C. 1994. Enlarged representative set of protein structures. Protein Science 3, 522.
4. Jin, Y., Pernice, M., and Boyd, R. H. 1996. MD simulations of bulk polymers beyond 10 ns using parallel computation on a workstation cluster. Computational and Theoretical Polymer Science 6, 9-13.
5. Nelson, M. T., Humphrey, W., Gursoy, A., Dalke, A., Kale, L. V., Skeel, R. D., and Schulten, K. 1996. NAMD: A parallel, object oriented molecular dynamics program. International Journal of Supercomputer Applications and High Performance Computing 10, 251-268.
6. Taylor, V. E., Stevens, R. L., and Arnold, K. E. 1997. Parallel molecular dynamics: Implications for massively parallel machines. Journal of Parallel and Distributed Computing 45, 166-175.
7. Zaloj, V., and Eber, R. 2000. Parallel computations of molecular dynamics trajectories using the stochastic path approach. Computer Physics Communications 128, 118-127.
8. Bywater, R., Gehring, J., Reinfeld, A., Rippmann, F., and Weber, A. 1999. Metacomputing in practice: a distributed compute server for pharmaceutical industry. Future Generation Computer Systems 15, 769-785.
9. Yap, T. K., Frieder, O., and Martino, R. L. 1998. Parallel computation in biological sequence analysis. IEEE Transactions on Parallel and Distributed Systems 9, 283-294.
10. Wei, L. and Altman, R. B. 1998. Recognizing protein binding sites using statistical descriptions of their 3D environments. In Proc. of the Pacific Symposium on Biocomputing, 497-508.

11. Bagley, S. C., and Altman, R. B. 1995. Characterizing the microenvironment surrounding protein sites. Protein Sciences 4, 622-635.
12. Bagley, S. C., and Altman, R. B. 1996. Conserved features in the active site of nonhomologous serine proteases. Folding and Design 1, 371-379.
13. Wei, L., Huang, E. S., and Altman, R. B. 1999. Are Predicted Structures Good Enough to Preserve Functional Sites? Structure 7, 643-650.
14. Foster, I., and Kesselman, C. 1997. Globus: A Metacomputing Infrastructure Toolkit. Intl J. Supercomputer Applications 11, 115—128.
15. Van Steen, M., Homburg, P., and Tanenbaum, A. S. 1997. The architectural design of Globe: A wide-area distributed system. Internal Report IR-422, Vrije Universiteit.
16. Litzkow, M., Livny, M., and Mutka, M. W. 1988. Condor – A hunter of idle workstations. Proc. of the 8th Int'l Conf. of Distributed Computing Systems, 104-111.
17. Gehring, J., and Streit, A. 2000. The MOL-kernel – A platform for multiform metacomputing services. 1st European GRID Forum Workshop.
18. Grimshaw, A. S., Nguyen-Tuong, A., and Wulf, W. A. 1995. Campus-wide computing: results using LEGION at the University of Virginia. UVaCS Technical Report CS-95-19.
19. Grimshaw, A. S., Wulf, W. A., and the LEGION team. 1997. The LEGION vision of a worldwide virtual computer. Communications of the ACM 40.
20. Wei, L. 1999. Automated annotation of sites in protein structures. PhD Dissertation, Medical Information Sciences, Stanford University.
21. Kabsch, W., and Sander, C. 1983. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 22, 2577-637.
22. Laskowski, R. A., Hutchinson, E. G., Michie, A. D., Wallace, A. C., Jones, M. L., and Thornton, J. M. 1997. PDBsum: A Web-based database of summaries and analyses of all PDB structures. Trends Biochem. Sci. 22, 488-490.
23. Baru, C., Moore, R., Rajasekar, A., and Wan, M. 1998. The SDSC Storage Resource Broker. In Procs. of CASCON '98, Toronto, Canada.
24. Blanchard, H., Grochulski, P., Li, Y., Simon, J., Arthur, C., Davies, P., Elce, J. S., and Cygler, M. 1997. Structure of a calpain Ca^{2+}-binding domain reveals a novel EF-hand and Ca^{2+}-induced conformational changes. Nature Structural Biology 4, 532-538.
25. Hasson, M. S., Schlicchting, I., Moulai, J., Taylor, K., Barrett, W., Kenyon, G. L., Babbitt, P. C., Gerlt, J. A., Petsko, G. A., and Ringe, D. 1998. Evolution of an enzyme active site: The structure of a new crystal form of muconate lactonizing enzyme compared with mandelate racemase and enolase. Proc. Natl. Acad. Sci. 95, 10396-10401.
26. Helin, S., Kahn, P. C., Guha, B. L., Mallows, D. G., and Goldman, A. 1995. The refined x-ray structure of muconate lactonizing enzyme from pseudomonas putida PRS2000 at 1.85 Å resolution. J. Mol. Biol. 254, 918-941.
27. Benson, S. D., Bamford, J. K. H., Bamford, D. H., and Burnett, R. M. 1999. Viral evolution revealed by bacteriophage PRD1 and human adenovirus coat protein structures. Cell 98, 825-833.
28. Athappilly, F. K., Murali, R., Rux, J. J., Cai. Z. and Burnett, R. M. 1994. The refined crystal structure of hexon, the major coat protein of adenovirus type 2, at 2.9 Å resolution. J. Mol. Biol. 242, 430-455.
29. Chen, C. C., Smith, T. J., Kapadia, G., Wasch, S., Zawadzke, L. E., Coulson, A., and Herzberg, O. 1996. Structure and kinetics of the ß–lactamase mutants S70A and K73H from staphylococcus aureus PC1. Biochemistry 35, 12251-12258.
30. Herzberg, O. 1991. Refined crystal structure of ß–lactamase from staphylococcus aureus PC1 at 2.0 Å resolution. J. Mol. Biol. 217, 701-719.