TEXTAL™: Automated Crystallographic Protein Structure Determination

Kreshna Gopal[1], Tod Romo[2], Erik Mckee[1], Kevin Childs[1], Lalji Kanbi[2], Reetal Pai[1], Jacob Smith[1], James Sacchettini[2] and Thomas Ioerger[1]

[1]Department of Computer Science, Texas A&M University, College Station, TX 77843-3112, USA

[1]{kgopal, emckee, kchilds, reetalp, thechao, ioerger}@cs.tamu.edu

[2]Department of Biochemistry and Biophysics, Texas A&M University, College Station, TX 77843-2128, USA

[2]{tromo, kanbi, sacchett}@tamu.edu

Abstract

This paper reports on TEXTAL™, a deployed application that uses a variety of AI techniques to automate the process of determining the 3D structure of proteins by x-ray crystallography. The TEXTAL™ project was initiated in 1998, and the application is currently deployed in three ways: (1) a web-based interface called WebTex, operational since June 2002; (2) as the automated model-building component of an integrated crystallography software called PHENIX, first released in July 2003; (3) binary distributions, available since September 2004. TEXTAL™ and its sub-components are currently being used by crystallographers around the world, both in the industry and in academia. TEXTAL™ saves up to weeks of effort typically required to determine the structure of one protein; the system has proven to be particularly helpful when the quality of the data is poor, which is very often the case. Automated protein modeling systems like TEXTAL™ are critical to the structural genomics initiative, a worldwide effort to determine the 3D structure of all proteins in a high-throughput mode, thereby keeping up with the rapid growth of genomic sequence databases.

Introduction

The aim of structural genomics is to quickly determine and characterize the 3D structure of all proteins and other macromolecules in nature (Burley et al. 1999), thus shedding light on their functions and enabling structure-based drug discovery. Recent years have witnessed a surge of effort towards high-throughput methods for protein structure determination. There is a pressing need for faster structure determination (Orengo et al. 1999), motivated by the exponential growth in the number of genomic sequences that are being uncovered, which require the knowledge of 3D structures for elucidating the functional significance of gene coding sequences.

In particular, there has been a growing demand for superior experimental and computational crystallography methods (Adams et al. 2004). There are many challenges at all stages of high-throughput protein crystallography, from data collection through structure solution, model-building and refinement to analysis. There has been significant progress in many of these stages (Hendrickson and Ogata 1997; Terwilliger and Berendzen 1999). Automated model-building, however, is one of the most difficult steps to improve, especially if the data collected is of poor quality. But automation of this bottleneck step is essential for high-throughput protein crystallography, for it takes days to weeks of tedious effort for a crystallographer to build a structure, even with high quality data (Kleywegt and Jones 1997).

TEXTAL™ undertakes the challenging task of automating protein structure determination, even with noisy data. This is achieved by using AI and pattern recognition techniques that try to mimic the human expert's decision-making processes. This paper discusses the significance, development, deployment, use and payoffs of TEXTAL™. The rest of the paper is organized in sections that discuss the following: (1) issues and challenges of automated protein model-building, and the motivation for AI-based approaches; (2) AI and pattern recognition techniques employed in TEXTAL™; (3) architecture of TEXTAL™ and its sub-systems; (4) deployment issues, use of the application, and the current as well as anticipated benefits; (5) development and maintenance of the system; (6) lessons learned and conclusion.

Problem Description

There are many steps involved in x-ray protein crystallography: first the protein has to be purified, and then a crystal has to be grown (the crystal is typically small, fragile, and usually contains 30-60% water). When x-rays are shone through the crystal, atoms of the crystal diffract the x-rays in a certain pattern, depending on the arrangement of the atoms. The intensities of the diffracted rays are determined, and an *electron density map* is produced by the Fourier transformation of the diffraction pattern. The electron density map can be viewed as an image of the electron cloud surrounding the protein

molecule. Interpreting a map essentially means building a model that fits this image.

One central problem is that the diffraction pattern contains information only about the intensities of diffracted waves; the phase information, which is also required for inferring the structure, is lost and has to be approximated by other methods. This is known as the *phase problem*. Furthermore, the sample of points at which intensities can be collected is limited, which constrains the degree to which atoms can be distinguished from one another. This imposes limits on the *resolution* of the map, measured in Angstroms (Å), where 1 Å = 10^{-10}m.

Solving the structure essentially means fitting the correct *amino acids* or *residues* into the density patterns in the right orientation (a protein is a sequence of typically hundreds of amino acids, or a *polypeptide* chain, that folds into a unique structure). A partially solved structure can be used to obtain better phase information, generate a superior map, which can then be re-interpreted. This process can go through several cycles, and it may take weeks, or even months of effort for an expert crystallographer to interpret a map, even with the help of molecular visualization programs. Protein structure determination can be laborious and inaccurate, depending on factors like the size of the structure, resolution of the data, etc. There are many sources of errors and noise, which distort the electron density map, making interpretation difficult (Richardson and Richardson 1985). Model building can also be subjective (Mowbray et al. 1999), where decisions are often based on what seems most reasonable in specific situations, with oftentimes little scope for generalization.

There are several methods that have been proposed for automated model-building: integrating modeling with phase refinement in ARP/wARP (Perrakis et al. 1999), expert systems (Feigenbaum, Engelmore, and Johnson 1997; Terry 1983), molecular-scene analysis (Glasgow, Fortier, and Allen 1993), database search (Jones and Thirup 1986; Diller et al. 1999; Holm and Sander 1991), using templates from the Protein Data Bank (Jones, Zou, and Cowtan 1991), template convolution and other FFT-based approaches (Kleywegt and Jones 1997), maximum-likelihood density modification (Terwilliger 2000), heuristic approaches to optimize fit into the density (Levitt 2001; Turk 2001), etc.

Many of these approaches depend on user-intervention and/or high quality data. In contrast, TEXTAL™ has been designed to be fully automated, and to work with average and even low quality data (around 2.8Å resolution); most maps are, in fact, noisy (Jones and Kjeldgaard 1997) due to difficulties in protein crystallization and other limitations of the data collection methods. In (Morris 2004), some of the popular model building packages, including TEXTAL™, are reviewed and compared.

Uses of AI Technology

AI and pattern recognition approaches are well-suited to address the various challenges of automated map interpretation: heavy dependency on expert knowledge in recognition of density patterns; decision-making at every step of the interpretation process, involving frequent backtracking; availability of growing databases of solutions which can be exploited by case-based reasoning.

TEXTAL™ has been designed to mimic a three-stage process that crystallographers typically employ:

(1) determine the "backbone" of the protein, which is essentially a main chain that contains central carbon atoms (known as Cα's) of all the amino acids that make up the protein;

(2) determine the identity and orientation of side chains or residues, based on knowledge of the positions of Cα's;

(3) post-processing routines to refine and enhance the model.

In the next sections, we describe the specific ways in which AI techniques are used in TEXTAL™. We emphasize that many of the novel AI techniques developed are potentially applicable to and useful for many other difficult problems, especially those which share common challenges with TEXTAL™: noisy and high-dimensional data, recognition of patterns in 3D, computationally costly retrieval from large databases, expensive domain expertise, etc.

Neural Network to Determine Positions of Cα atoms

To determine the 3D coordinates of Cα atoms, TEXTAL™ uses a feed-forward neural network to predict the distance of many candidate positions (chosen along the medial axis of the electron density map), and selects the ones that are deemed closest to true Cα's, factoring in domain knowledge, especially about constraints on distances between Cα's. The objective of the neural network is to learn the relationship between characteristics of electron density patterns around a point and its proximity to Cα atoms. Thirty eight numeric features are used to characterize the local density; these features are fed to the network, which uses one layer of 20 hidden units with sigmoid thresholds, and outputs the predicted distance to a true Cα atom. The network is trained with a set of points in maps of solved proteins, with known distances to true Cα's, and the network weights are optimized using *backpropagation*. For more details, refer to (Ioerger and Sacchettini 2002).

Heuristic Search to Build Chains

An AI-based approach is also used to link the Cα atoms (as predicted by the neural network described earlier) into backbone chains, based on how well the chains built fit typical structural motifs in proteins, known as *secondary structures*. Linking Cα atoms into chains is a combinatorial search problem; whenever possible, an exhaustive search is done to create an optimum solution. When a complete search becomes intractable, TEXTAL™ uses heuristics to choose between various options for chains, based on criteria that favor better adherence to

stereo-chemical constraints and secondary structures that occur commonly in proteins. Essentially, these heuristics and decision criteria try to capture the type of reasoning that experienced crystallographers employ. It should be emphasized that automation of this deliberation is particularly challenging because noisy data (like incorrect or missing connections in the density patterns) can be easily misleading, which can potentially lead to a succession of wrong decisions. A thorough discussion of the methods used to build the backbone can be found in (Ioerger and Sacchettini 2002).

Case-Based Reasoning to Stitch Chains

This is a backbone refinement step that follows the initial construction of the backbone chains; it attempts at slightly modifying and connecting different chains together, especially in regions where the backbone makes a loop. A case-based reasoning approach that bears resemblance to a method proposed by (Jones and Thirup 1986) is employed to "stitch" chains - regions of the structure that probably should have been connected (typically at close extremities of different chains) are identified and a database of solved density patterns (constructed from ~100 maps) is searched to find the most similar case. The case matching is done by superposing all chain fragments (of 7 to 11 consecutive Cα atoms) from the database with the region under consideration, and computing the root mean square deviation. If the deviation is small enough, and the electron density in the region is adequately high, then stitching is justified, which may entail adding new Cα atoms. This approach tries to circumvent the problem of noisy electron density data, especially regions with broken density.

Case-Based Reasoning and Nearest Neighbor Learning to Model Side Chains

After the backbone is built and refined, the density is fitted with side chains or amino acids by using the information on Cα positions. Spherical regions with a radius of 5Å are defined around the Cα atoms, and for each region, a database of ~50,000 previously solved regions [constructed from maps of ~200 proteins from PDBSelect (Hobohm et al. 1992)] is searched to identify the best match. Essentially this involves recognition of the unknown patterns of density by comparison to other known cases. This can be achieved by using a similarity metric based on how well the density distribution of the two regions superimpose over each other. But such an objective similarity metric involves computing the optimal superposition between two 3D regions. Since the number of possible 3D orientations of a region is very large, this metric is expensive and we cannot afford to run it on the whole database. Thus, we use an inexpensive and approximate feature-based measure of similarity to filter k (400, for instance) cases, based on k-Nearest Neighbor learning; the selected cases are then examined by the more expensive metric to make the final choice.

There are two noteworthy issues related to this approach: (1) a fast and effective similarity metric has to be defined to do the filtering, such that as many good matches as possible are caught in the top k cases filtered. In (Gopal et al. 2004b), we compare various similarity measures, and argue that probabilistic and statistical measures outperform geometric ones (like those based on Manhattan or Euclidean distance); (2) the choice of k is important since it influences the performance, both in terms of computational cost and quality of retrievals. In (Gopal et al. 2004a), we empirically and theoretically analyze the choice of a suitable k, and provide a model to predict k based on a loss function that represents the ability of approximate measures of similarity to rank good matches (according to the objective metric) as best possible. An in-depth discussion on the side chain modeling system as a whole can be found in (Ioerger and Sacchettini 2003).

Feature Extraction and Weighting

The inexpensive measure of similarity that we use to compare two density regions (for side chain modeling) is based on numeric features that characterize density patterns. Constructing and selecting features can be challenging (Aha 1998; Liu and Motoda 1998; Ioerger 1999), especially since our features are in 3D and they have to be rotation-invariant (since the regions that we want to compare using the features can occur in any 3D orientation). Seventy six features have been chosen and categorized into four classes that capture different types of information about density patterns: statistical features related to the electron density distribution, information on symmetry (as defined by the distance from the center of the region to its center of mass), moments of inertia (and their ratios), and features that try to represent the geometric shape of the region.

Various features contribute differently to the description of regions; thus features have to be weighted accordingly. We use a feature weighting strategy called SLIDER, which adjusts weights incrementally, such that, for a given set of regions, known matching regions are better ranked than known mismatching ones. SLIDER employs a *filter* approach (Kira and Rendell 1992) to feature weighting. One salient innovation in our method is a more efficient and informed way of finding the weight values that are most promising candidates for update, thereby circumventing the intractability of exhaustive search over all possible weight vectors. In fact, SLIDER uses a greedy, heuristic-based approach, where in each iteration we consider only those weights at which matches and mismatches switch as nearer neighbors to query instances; these weights can be efficiently computed. The classification accuracy is more likely to change at these particular weights (which we refer to as *crossovers*), thereby making the search fast and effective. For more details, refer to (Gopal et al. 2005).

Detection of Disulfide Bridges through Linear Discriminant Analysis

A disulfide bridge is a covalent bond between the sulfur atoms of two *cysteine* residues from one or neighboring polypeptide chains. The residues with disulfide bridges can be located anywhere in the chain, and thus they contribute to the stability of the protein. Disulfide bridges occur in roughly one out of every four proteins; localizing them in an electron density map will help the crystallographer in modeling the backbone as well as side chains, especially since the presence of a disulfide bridge will reveal the position of cysteine residues.

Disulfide bridges are detected by the following method: First, local spherical regions in the electron density map are characterized by 76 numeric features (the same ones that are used for the side chain modeling step described earlier). Then a linear discriminant model is applied to estimate resemblance to a disulfide bridge, based on a training experience with known cases from a disulfide class and a non-disulfide class of examples. The training cases are used to optimize the parameters of the linear discriminant. In particular, the Fisher linear discriminant model (Fisher 1936) is used to optimally maximize class separation, and minimizing variance within each class. This classification method projects the high-dimensional data onto an optimal line in space, along which classification is performed, using a single threshold to distinguish between the two classes. A detailed discussion on this method can be found in (Ioerger 2005).

Application Description

In this section, we briefly describe the architecture of TEXTAL™, which combines both AI and non-AI techniques to address the various facets of the complex problem of protein structure determination. TEXTAL™ has been designed to be modular, where different components can be used independently or in various possible combinations. TEXTAL™ uses two standard formats of data as input/output: electron density maps are in XPLOR format (Brünger 1992); atoms and their coordinates (to represent chains, partially or completely solved models) are in PDB format (Berman et al. 2000).

TEXTAL™ is made up of three major sub-systems (as shown in Fig. 1):

(1) CAPRA, or C-Alpha Pattern Recognition Algorithm, models the backbone (or main chain) of the protein. Essentially it takes an electron density map as input, and outputs a PDB file containing a set of Cα chains representing the true backbone as best possible. CAPRA is made up of several modules, as described in Fig. 1.

(2) LOOKUP uses the output of CAPRA to model the residues (or side chains), using case-based reasoning and nearest neighbor learning. Essentially, LOOKUP takes spherical regions (of 5Å radius) around the Cα atoms determined in CAPRA, and retrieves their best matches from a database of solved cases. The known structures of the matches are used to model the side chains in a piecewise manner.

(3) POST-PROCESSING routines refine the initial model built by LOOKUP. Two main routines in this sub-system are:

- *sequence alignment*, where the sequence of residues in the initial model produced by LOOKUP is aligned with the known sequence of amino acids of the protein (Smith and Waterman 1981). This enables another round of LOOKUP to make corrections in the amino acid identities initially determined.
- *real space refinement*, where slight adjustments in the positions of atoms are made to better fit the density (Diamond 1971).

TEXTAL™ usually produces a very reasonable first model of the protein, saving the crystallographer a lot of time. The model produced by TEXTAL™ can be manually improved, or used to generate better phase information and create a better electron density map, which can fed back to TEXTAL™.

The performance of TEXTAL™ depends on the size and complexity of the model, and the quality of the data. TEXTAL™ and its sub-systems have been designed to work for a widely variety of proteins, of different sizes, with different structural components. TEXTAL™ usually outputs a reasonable model even with poor quality data (i.e. around 3Å resolution). Typically CAPRA builds about 90% of the backbone, with less than 1Å root mean square distance error. TEXTAL™ usually models about 50% of the side chains with the correct identity. In cases where TEXTAL™ cannot find the exact amino acid, it typically places one that is structurally similar to the correct one. For a more detailed appraisal of the quality of modeling done by CAPRA and TEXTAL™, refer to (Ioerger and Sacchettini 2002; Ioerger and Sacchettini 2003; Holton et al. 2000; Gopal et al. 2003).

Deployment, Use and Payoff

Deployment through WebTex

The first version of WebTex, the web-interface of TEXTAL™ (http://textal.tamu.edu:12321) was made available to the public in June 2002. Users have to register online for an account, and on approval, they can upload their electron density maps, specify options through a simple interface, and submit their jobs. These are processed on our server (on an SGI Origin 2000) at Texas A&M University, and the results are automatically emailed to users. Typically it takes a couple of hours to run TEXTAL™ on a medium-sized protein, and around 20 minutes to run CAPRA. Users can also monitor online and in real-time the progress of their jobs, and view and download all the data files related to each of their runs. Unsuccessful runs are automatically detected and the user as well as the systems administrator are informed through

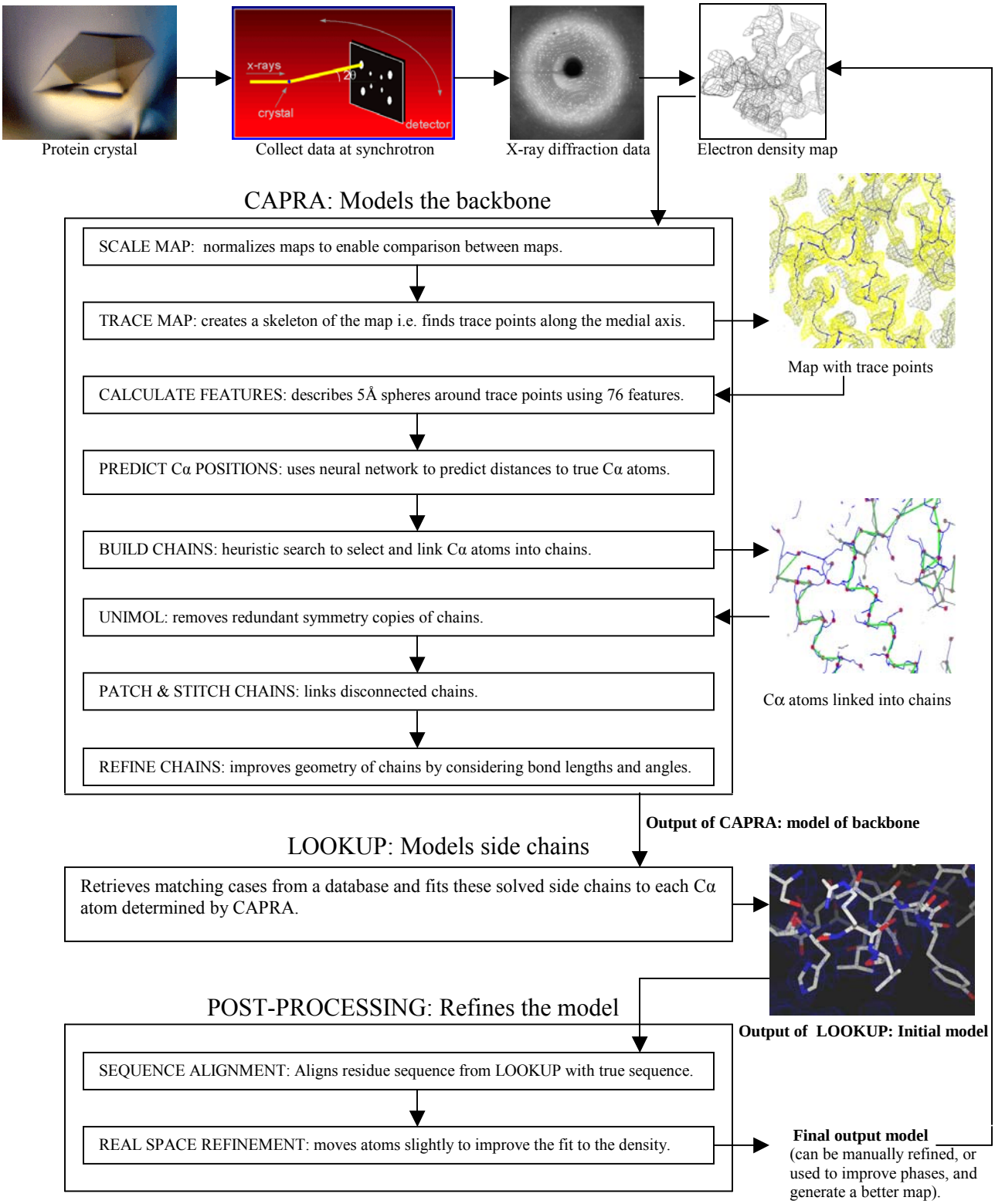

Fig. 1. Architecture of the TEXTAL™ system, showing the three main sub-systems: CAPRA, LOOKUP and POST-PROCESSING. Some modules, like UNIMOL and SEQUENCE ALIGNMENT, are optional. Several utilities (such as one to find disulfide bridges) that can be run independently of the flow displayed are not shown.

system generated emails – unsuccessful runs are rare, and inevitably occur for reasons like input data uploaded is in the wrong format, input files are too large, etc.

WebTex is freely available to non-profit users, but access has been granted only to those who share membership with the TEXTAL™ group in various structural genomics consortia. Limited access to WebTex has been necessary because TEXTAL™ is a computationally intensive system (LOOKUP, in particular). In fact, restrictions are imposed on the size of maps that can be uploaded, and on the number of concurrent jobs that can be submitted. Currently CAPRA jobs can be submitted without an account, since it is relatively inexpensive. Another practical consideration is our obligation to maintain confidentiality of users' data, and reassure users about the same. During the period from June 2002 to March 2005, 403 jobs have been submitted and successfully processed on WebTex. These jobs have been submitted by 114 users from 65 institutions (both academic and industrial) in 18 countries.

The payoff of TEXTAL™ is mostly in terms of time saved to solve a structure; while a crystallographer may spend several days and sometimes weeks of painstaking effort for interpreting one map, TEXTAL™ produces a solution in a couple of hours, without human intervention. Even if the model produced by TEXTAL™ is only partially accurate, it provides a reasonable initial solution, which can be manually refined by the crystallographer to produce a more complete model. The benefits are hard to quantify, since they vary largely with the size and quality of maps, and they depend on the crystallographer working on the map. But the benefits of TEXTAL™ to users are evident, as suggested by the consistency with which maps are submitted to the WebTex site; the maximum number of maps that have been submitted by a single user is currently 30 (submitted over a span of about 2 years).

Deployment through PHENIX

PHENIX (Python-based Hierarchical ENvironment for Integrated Xtallography) is an international initiative to develop a software package for automated x-ray crystal structure determination, especially at medium to low resolution (Adams et al. 2004; http://www.phenix-online.org). The PHENIX software provides a variety of algorithms to proceed from reduced intensity data to a refined molecular model, and facilitate structure solution for both the novice and expert crystallographer. The architecture of the PHENIX system is depicted in Fig. 2. The Python scripting language (http://www.python.org) provides the backbone of the system. The Boost.Python library (Abrahams and Grosse-Kunstleve 2003) is used to integrate C++ code into Python. On top of this, the data objects, crystallographic tasks, strategies (or network of tasks), and finally a graphical user interface are constructed. The Project Data Storage makes use of the pickle mechanism in Python to store data on the file system.

The main components and developers of the PHENIX system are:

- CCTBX: The Computational Crystallography Toolbox provides a suite of programs for high-throughput structure determination, implemented at the Lawrence Berkeley National Laboratory (Adams et al. 2003; http://cci.lbl.gov/index.html). The developers of CCTBX have also been involved in the design and implementation of the underlying architecture and user-interface of the PHENIX software.
- PHASER: A program for phasing macromolecular crystal structures using maximum likelihood methods, developed at the University of Cambridge (http://www-ructmed.cimr.ac.uk/phaser/index.html; Read 2001).
- SOLVE and RESOLVE: These systems are being developed at Los Alamos National Laboratory (http://www.lanl.gov). SOLVE aims at automated crystallographic structure solution (Terwilliger and Berendzen 1999), and RESOLVE performs statistical density modification, local pattern matching, automated model building, and prime-and-switch minimum bias phasing (Terwilliger 2000; Terwilliger 2002).
- TEXTAL™: The automated electron density map interpretation component, developed at Texas A&M University (http://textal.tamu.edu:12321; Ioerger and Sacchettini 2003).

The PHENIX industrial consortium, which consists of commercial organizations that actively collaborate with the development groups, currently include Chiron Corp., Genentech Inc., Glaxo-Smith-Kline, Plexxikon Inc., Wyeth Ayerst Research.

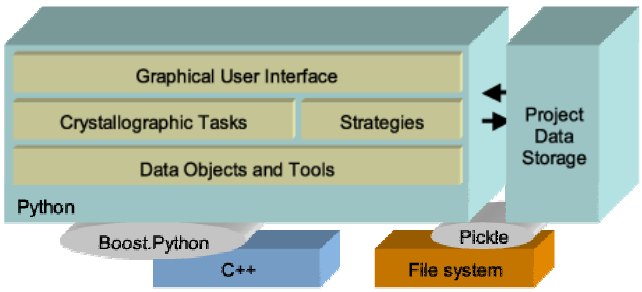

Fig. 2. Architecture of the PHENIX system.

The first alpha test version of PHENIX was released in July 2003, mainly to consortium members and selected users. Six more releases have been made since July 2003. The next release is scheduled for April 2005. The software is available for commonly used computing platforms: Redhat Linux, HP Tru64, SGI Irix 6.5 and currently under development, Windows and Macintosh OSX versions. PHENIX is a large-scale, ambitious project that is expected to have significant impact in the field of protein

crystallography. The main payoff is the availability of a wide and comprehensive range of high-throughput crystallography tools in an integrated computational environment. Researchers benefit substantially from the ease and flexibility to link various crystallographic tasks together, without having to resort to low-level programming.

Deployment through Binary Distributions

In September 2004, Linux and OSX versions of TEXTAL™ were made available for download from our website (http://textal.tamu.edu:12321) and on CD-ROM. TEXTAL™ site licenses can be procured from our website. License keys (based on MAC addresses of target machines) are automatically generated and emailed to applicants. The distributions of TEXTAL™ provide more flexibility to the user as compared to WebTex; it allows TEXTAL™ modules to be invoked from the command line as well as through a Tcl/Tk-based interface called WinTex, and provides for many options. Another major advantage of the binary distribution over WebTex is that the user does not need to be concerned about ensuring confidentiality of usually valuable x-ray diffraction data. Since the release of the distributions in September 2004, 31 TEXTAL™ licenses have been granted on a trial basis for a limited period.

Development and Maintenance

The TEXTAL™ project was initiated in 1998 as a collaboration between the departments of Computer Science and Biochemistry & Biophysics at Texas A&M University. Twenty researchers and programmers have so far been involved in the project, and the size of the TEXTAL™ staff averages to about 8. The TEXTAL™ software is about 100,000 lines of C/C++, Perl and Python code. The development platforms include various versions of Irix, Linux, Macintosh, and Windows. We use the Concurrent Versions Systems, or CVS (http://www/cvshome.org) to coordinate the development of TEXTAL™, including integration with the PHENIX system. CVS enables tracking of code updates, allows developers to access the latest version of the code from anywhere, and allows multiple developers to work simultaneously on the same code in safety.

Conclusion

TEXTAL™ is an excellent illustration of effective integration of AI technology with other tools to solve a real, significant and difficult problem in an inter-disciplinary fashion. In this paper, we have emphasized the importance and challenges of high-throughput protein crystallography in structural genomics, and the contribution of automated protein model-building systems like TEXTAL™. We described a variety of AI and pattern recognition techniques that were necessary to address the various facets of this complex problem: neural network, heuristic search, case-based reasoning, nearest neighbor learning, linear discriminant analysis, feature extraction and weighting. We argue that many of the AI issues dealt with, and techniques developed, can be used in other domains, typified by the need to recognize visual patterns (especially in 3D), noisy inputs, expensive and extensive domain knowledge encoded in growing databases, computationally costly case matching and retrieval. Furthermore, we alluded to many practical issues of deployment: maintaining a trade-off between accuracy and speed of modeling; multiple interfaces and modes of deployment to meet varying needs of users; secure system maintenance and integration, especially with distributed development; data confidentiality, license agreements and other legal issues; support for multiple platforms; etc.

TEXTAL™ is continuously being enhanced; existing modules are being improved, and new features added. Recent developments include: transformation of the skeleton (or trace) of a density map such that a symmetrically unique protein macromolecule is covered; identification of *non-crystallographic symmetry*; and simplex optimization (Nelder and Mead 1965) to improve modeling in LOOKUP.

References

Abrahams, D. and Grosse-Kunstleve, R.W. 2003. *C/C++ Users Journal* 21: 29-36.

Adams, P.D., Gopal, K., Grosse-Kunstleve, R.W., Hung, L.W., Ioerger, T.R., McCoy, A.J., Moriarty, N.W., Pai, R., Read, R.J., Romo, T.D., Sacchettini, J.C., Sauter, N.K., Storoni, L.C., and Terwilliger, T.C. 2004. Recent developments in the PHENIX software for automated crystallographic structure determination. *Journal of Synchrotron Rad.* 11: 53-55.

Adams, P.D., Grosse-Kunstleve, R.W., Hung, L.W., Ioerger, T.R., McCoy, A.J., Moriarty, N.W., Read, R.J., Sacchettini, J.C., and Terwilliger, T.C. 2002. PHENIX: Building new software for automated crystallographic structure determination. *Acta Cryst.* D58:1948-1954.

Aha, D.W. 1998. Feature weighting for lazy learning algorithms. In Liu H., and Motoda, H. eds. *Feature Extraction, Construction and Selection: A Data Mining Perspective.* Boston, MA: Kluwer.

Berman H.M., Westbrook J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E. 1992. The Protein Data Bank. *Nucleic Acids Research* 28: 235-242.

Brünger, A.T. 1992. XPLOR manual, version 3.1, Yale University, New Haven, CT.

Burley, S.K., Almo, S.C., Bonanno, J.B., Capel, M., Chance, M.R., Gaasterland, T., Lin, D., Sali, A., Studier, W., and Swaminathian, S. 1999. Structural genomics: beyond the human genome project. *Nature Genetics* 232:151-157.

Diamond, R. 1971. A real-space refinement procedure for proteins. *Acta Cryst*. A27:436-452.

Diller, D.J., Redinbo, M.R., Pohl, E., and Hol, W.G.J. 1999. A database method for automated map interpretation in protein crystallography. *PROTEINS: Structure, Function, and Genetics* 36:526-541.

Feigenbaum, E.A., Engelmore, R.S., and Johnson, C.K. 1997. A correlation between crystallographic computing and artificial intelligence research. *Acta Cryst*. A33:13-18.

Fisher, R.A. 1936. The use of multiple measurements in taxonomic problems. *Annals Eugenics* 7:179-188.

Glasgow, J., Fortier, S., and Allen, F. 1993. Molecular scene analysis: Crystal structure determination through imagery. In Hunter, L., ed. *Artificial Intelligence and Molecular Biology*. Cambridge, MA: MIT Press.

Gopal, K., Pai, R., Ioerger, T.R., Romo, T.D., and Sacchettini, J.C. 2003. TEXTAL™: Artificial intelligence techniques for automated protein structure determination. In *Proceedings of the Fifteenth Conference on Innovative Applications of Artificial Intelligence Conference*, 93-100.

Gopal, K., Romo, T.D, Sacchettini, J.C, and Ioerger, T.R. 2004a. Efficient retrieval of electron density patterns for modeling proteins by x-ray crytallography. In *Proceedings of the International Conference on Machine Learning and Applications*, 380-387.

Gopal, K., Romo, T.D, Sacchettini, J.C, and Ioerger, T.R. 2004b. Evaluation of geometric & probabilistic measures of similarity to retrieve electron density patterns for protein structure determination. In *Proceedings of the International Conference on Artificial Intelligence*, 427-432.

Gopal, K., Romo, T.D, Sacchettini, J.C, and Ioerger, T.R. 2005. Determining relevant features to recognize electron density patterns in x-ray protein crystallography. To appear in *Journal of Bioinformatics & Computational Biology*.

Hendrickson, W.A. and Ogata, C.M. 1997. Phase determination from multiwavelength anomalous diffraction measurements. *Methods in Enzymology* 276:494-523.

Hobohm, U., Scharf, M., Schneider, R., and Sander, C. 1992. Selection of a represetative set of structures from the Brookhaven Protein Data Bank. *Protein Science* 1: 409-417.

Holm, L. and Sander, C. 1991. Database algorithm for generating protein backbone and side-chain coordinates from a Cα trace. *J. Mol. Biol*. 218:183-194.

Holton, T.R., Christopher, J.A., Ioerger, T.R., and Sacchettini, J.C. 2000. Determining protein structure from electron density maps using pattern matching. *Acta Cryst*. D46:722-734.

Ioerger, T.R. 1999. Detecting feature interactions from accuracies of random feature subsets. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, 49-54.

Ioerger, T.R. 2005. Automated detection of disulfide bridges in electron density maps using linear discriminant analysis. *Journal of Applied Crystallography* 38(1):121-125.

Ioerger, T.R. and Sacchettini, J.C. 2002. Automatic modeling of protein backbones in electron-density maps via prediction of Cα coordinates. *Acta Cryst.* D5:2043-2054.

Ioerger, T.R and Sacchettini, J.C. 2003. The TEXTAL system: artificial intelligence techniques for automated protein model building, In Sweet, R.M. and Carter, C.W., eds. *Methods in Enzymology* 374: 244-270.

Jones, T.A. and Kjeldgaard, M. 1997. Electron density map interpretation. *Methods of Enzymology* 277: 173-208.

Jones, T.A. and Thirup, S. 1986. Using known substructures in protein model building and crystallography. *EMBO Journal* 5(4):819-822.

Jones, T.A., Zou, J.Y., and Cowtan, S.W. 1991. Improved methods for building models in electron density maps and the location of errors in these models. *Acta Cryst*. A47:110-119.

Kira, K. and Rendell, L.A. 1992. A practical approach to feature selection. In *Proceedings of the Ninth International Conference on Machine Learning*, 249-256.

Kleywegt, G.J. and Jones, T.A. 1997. Template convolution to enhance or detect structural features in macromolecular electron density maps. *Acta Cryst*. D53:179-185.

Levitt, D.G. 2001. A new software routine that automates the fitting of protein X-ray crystallographic electron density maps. *Acta Cryst*. D57:1013-1019.

Liu, H. and Motoda, H. eds. 1998. *Feature Extraction, Construction, and Selection: A Data Mining Perspective*. Boston, MA: Kluwer.

Morris, R.J. 2004. Statistical pattern recognition for macromolecular crystallographers. *Acta Cryst.* D60, 2133-2143.

Mowbray, S.L., Helgstrand, C., Sigrell, J.A., Cameron, A.D., and Jones, T.A. 1999. Errors and reproducibility in electron-density map interpretation. *Acta Cryst*. D55: 1309-1319.

Nelder, J. A. and Mead, R. 1965. A simplex method for function minimization. *Comput. J.* 7, 308-313.

Orengo, C.A., Pearl, F.M., Bray, J.E., Todd, A.E., Matin, A.C., Lo Conte, L., and Thornton, J.M. 1999. *Nucleic Acids Res.* 27: 275-279.

Perrakis, A., Morris, R., and Lamzin, V. 1999. Automated protein model-building combined with iterative structure refinement. *Nature Structural Biology* 6:458-463.

Read, R. 2001. Pushing the boundaries of molecular replacement with maximum likelihood. *Acta Cryst*. D57: 1373-1382.

Richardson, J.S. and Richardson, D.C. 1985. Interpretation of electron density maps. *Methods in Enzymology* 115:189-206.

Smith, T.F. and Waterman, M.S. 1981. Identification of common molecular subsequences. *J. Mol. Biol*. 147:195-197.

Terry, A. 1983. The CRYSALIS Project: Hierarchical Control of Production Systems. Technical Report HPP-83-19, Stanford University, Palo Alto, CA.

Terwilliger, T.C. 2000. Maximum-likelihood density modification. *Acta Cryst.* D56:965-972.

Terwilliger, T.C. 2002. Automated main-chain model-building by template-matching and iterative fragment extension. Acta Cryst. D59, 34-44.

Terwilliger, T.C. and J. Berendzen. 1999. Automated MAD and MIR structure solution. *Acta Cryst*. D55, 849-861.

Turk, D. 2001. Towards automatic macromolecular crystal structure determination. In Turk, D. and Johnson, L. eds. *Methods in Macromolecular Crystallography*. NATO Science Series I, vol. 325, 148-155.