

A Computational Pipeline for Protein Structure Prediction and Analysis at Genome Scale

Manesh Shah^{1,3}, Sergei Passovets^{1,3}, Dongsup Kim¹, Kyle Ellrott³, Li Wang^{1,3}, Inna Vokler^{1,3}, Philip LoCascio¹, Dong Xu^{1,3}, Ying Xu^{1,2,3,*}

¹Life Sciences Division and ²Computer Sciences and Mathematics Division, Oak Ridge National Laboratory, TN 37830-6480, USA

³School of Genome Science and Technology, University of Tennessee, Knoxville, TN 37922, USA

* Correspondence author: xyn@ornl.gov

Abstract

The tertiary (3D) structure of a protein contains the essential information for understanding the biological function of the protein at the molecular and cellular levels. Traditionally, protein 3D structures are solved using experimental techniques, like x-ray crystallography or nuclear magnetic resonance (NMR). While these experimental techniques have been the main workhorse for protein structure studies in the past few decades, it is becoming increasingly apparent that they alone cannot keep up with the production rate of protein sequences as a result of worldwide genome sequencing and bioinformatics efforts. Fortunately, computational techniques for protein structure predictions have matured to such a level that they can complement the existing experimental techniques. In this paper, we present an automated pipeline for protein structure prediction. The centerpiece of the pipeline is a threading-based protein structure prediction system, called PROSPECT, which we have been developing for the past few years. The pipeline consists of seven logical phases, utilizing a dozen tools: (1) preprocessing to identify protein domains in the input sequence, (2) compilation of functional and structural information about a target protein through database search, (3) protein triage to determine which process and prediction branches to use, (4) protein fold recognition for identification of native-like folds of a target protein, (5) protein structure prediction to generate atomic structure models, (6) quality assessment of predicted structure, and (7) prediction result validation. Different processing and prediction branches are determined automatically and employed for each individual protein, by a prediction pipeline manager, based on identified characteristics of the protein. The pipeline has been implemented to run in a heterogeneous computational environment, consisting of Alpha, Solaris and Linux servers, a 64-node Linux cluster and a wide range of ORNL supercomputers as a client/server system with a web interface. XML (Extensible Markup Language) has been extensively used for data representation and data exchange between different components of the pipeline. A number of genome-scale applications have been carried out on microbial genomes. Here we present one genome-scale application on *Caenorhabditis elegans*.

1. INTRODUCTION

Since the inception of human genome sequencing project (Lander *et al.*, 2001; Venter *et al.*, 2001), over 100 genomes have been sequenced and their genes predicted. It is expected that over 1,000 genomes will be sequenced and their genes predicted, within the next ten years. A major challenge facing the computational and experimental biologists is how to derive the biological functions of these genes at a comparable pace. BLAST/PSI-BLAST (Altschul *et al.*, 1997) have been one of the key tools for inferring biological functions of unknown genes through sequence-based homology search. Though highly effective, the limitation of this method is also clear. The general observation has been that about 30-40% of genes in a newly sequenced genome do not

have any recognizable functional assignments at the molecular or cellular level, using sequence-based approaches like BLAST (Altschul et al., 1990). In addition, sequence-based homology search methods can provide functional information only at a low-resolution level – little can be inferred, from this information, about the mechanism of how a gene’s biological function is realized. Structure-based homology search methods can clearly provide additional information for the functional inference of proteins, as demonstrated in the recent CASP contests (CASP, 1993, 1995, 1997, 2001) and through a large number of real life applications (Xu et al., 2001). Firstly, structure-based homology search uses both the sequence information of a gene (or its protein product) and the structure information of the protein, making it generally more sensitive for remote homology identification. Secondly, the predicted protein structure could provide clues about the functional mechanism of a protein product. In addition to being useful to large-scale genome annotation projects, structure prediction techniques are gradually becoming part of the standard toolkit of protein biochemists and molecular biologists for their protein studies. Such prediction capabilities can be used to quickly generate structural/functional hypotheses to guide the design of experiments for validation and testing. The effectiveness of such integrated computational and experimental protocols has been demonstrated by numerous applications (Xu et al., 2001).

Protein structure prediction is a complex multi-faceted process. Different classes of proteins, say soluble *versus* membrane-associated proteins or proteins with or without structural homologues, may require different computational techniques for their structure predictions, due to their different physicochemical or other properties. A protein could have multiple structural domains, and prediction of the whole protein structure with multiple domains could prove to be computationally intractable or there may not be a structural template for the whole protein, at least at the current stage. An observation is that the folding of each structural domain of a protein, to a large degree, occurs independently of other domains, and hence each domain structure can be predicted independently. The problem then becomes how to identify the domain boundaries in a protein sequence. Some protein sequence may contain additional peptides, like signal peptides, which will not be involved in the folding process into its native structural conformation and will eventually be cleaved out. Just to name a few things that can complicate a structure prediction process.

Currently, there are a large number of computational tools available on the Internet, for prediction and characterization of different aspects of protein structures. Each of these prediction tools is designed to solve a particular range of structure prediction/characterization problems. The selection and application of these tools require special training and knowledge about individual tools. Interpretation of a prediction result could represent an even more challenging problem. It will generally require in-depth knowledge about the prediction tools and general knowledge and understanding about protein structures – to an expert’s eyes, some predicted structures simply do not fit the general “profile” of protein structures (Murzin et al, 1995). Numerous questions could be asked about a predicted structure. Is the overall structural fold correct of a predicted structure? Is the whole backbone structure or part of the backbone reliable, knowing that the fold is correct? How to determine which parts of a predicted structure are reliable and at what level of accuracy? This may often require running different (independent) prediction tools, and use their prediction results to validate each other. For an experienced structure predictor, it may take days (if not longer) to make a sensible structure prediction (if predictable using the existing tools) as this process will involve running different tools, possibly multiple times using different sets of parameters, and cross-validation through comparing different prediction results. A typical bench biologist may be shy away from this lengthy and complicated prediction process, which clearly requires learning and in-depth understanding of different prediction tools.

It will be a great service to the molecular biology community if a seamless computational procedure can be built to automate the application of appropriate prediction tools and interpretation of the prediction results. Such a procedure will involve (a) the use of relevant prediction tools for different prediction scenarios; (b) a capability to triage the prediction targets and automatically determine the prediction pathway and select a set of prediction tools for each protein, (c) seamless integration of these tools through automated data conversion, parameter selection and procedure calling, (d) incorporation of expert predictors' knowledge to guide the automated prediction process; and (e) capability for interpretation of prediction results.

Several prediction pipelines have been developed for genome-scale protein structure predictions. GeneAtlas (Kitson et al., 2002) is one such commercial software product. It uses PSI-BLAST (Altschul et al. 1997) for sequence comparison, SeqFold (<http://www.accelrys.com/insight/seqfold.html>) for fold recognition, and MODELLER (Sali and Blundell, 1993) for detailed 3D-structure model construction. It uses Profile-3D (http://www.accelrys.com/insight/Profiles-3D_page.html) for quality assessment of each predicted structure. Another pipeline for large-scale protein structure prediction, using comparative approaches, is ModPipe (Sanchez and Sali 1998). It uses a set of prediction tools similar to those used in GeneAtlas. The quality of each predicted structure is assessed using a statistical energy function, sequence similarity with the modeling template, and a measure of structural compactness (Sanchez and Sali 1998). The PAT database (<http://arabidopsis.sdsc.edu/>) contains predicted protein structures encoded by the *Arabidopsis thaliana* genome. These structures are predicted using a prediction pipeline at the San Diego Supercomputing Center, including somewhat expanded list of prediction tools, compared to the above two pipelines. In addition to the tools used in GeneAtlas, ModPipe also uses COILS (Lupas et al. 1991) for coiled-coil prediction, TMHMM (Krogh et al. 2001) for transmembrane protein identification, and SignalP (Nielsen et al., 1997) for signal peptide prediction. The reliability of each method is assessed and incorporated into the pipeline, and the final predictions are ranked according to their reliability scores. Unfortunately, none of these prediction pipelines are freely available for use by the research community. What they generally provide is a searchable database of predicted protein structures that have been generated by their respective prediction pipelines.

We have recently developed a computational pipeline for large-scale protein structure predictions. A key distinguishing feature of our system from the aforementioned ones, is that we have incorporated a great deal of the knowledge of expert human predictors into the pipeline, while other prediction pipelines have simply "pipelined" a collection of prediction tools in a simple minded manner. As noted by M. Sippl (oral presentation at CASP4 meeting, 2000), one of the key reasons that computer-assisted human predictors have outperformed automated computer predictions in CASP is that human predictors can refine computer predictions through a better interpretation of the prediction results, integration of additional structural and functional information into the prediction process in an iterative manner, cross-validation of prediction results from different tools, and application of human intuition and judgement. During previous CASPs, we have developed an effective computer-assisted manual prediction procedure (Xu et al, 1999, Xu et al, 2001), which involves a set of (human) decision making and inference processes. These include tools selection criteria for specific conditions, integration of information from different sources, cross-validation of prediction results from different tools and intelligent interpretation of prediction results. A significant portion of this manual process has been incorporated into our prediction pipeline. Another unique feature of our prediction pipeline is that it is accessible to the research community over the Internet (<http://compbio.ornl.gov/proteinpipeline/>). This is possible, in large part, due to the availability of powerful supercomputing resources available to us at the Oak Ridge National Laboratory. The pipeline can be accessed through a web interface, facilitating interactive communication between the pipeline and the user.

2. CONCEPTUAL DESIGN OF THE PIPELINE

Among the three most prevalent techniques (*ab initio* folding, protein threading and homology modeling) for protein structure prediction, *protein threading* (Bowie et al., 1991; Jones et al., 1992; Sippl et al., 1992) represents probably the most generally applicable and reliable approach. Protein threading predicts the backbone structure of a target protein through identifying its native-like structural folds from a protein structure database like PDB (Westbrook et al., 2000), and finding the energetically most “favorable” placement (or alignment) of the sequence onto each of the identified structural folds. It is estimated that protein threading, in theory, is applicable to 60-70% of soluble proteins for their fold recognition and backbone structure prediction (Montelione and Anderson, 1999). Though effective, structure prediction by protein threading often requires human involvement in pre-processing the target sequence and collecting additional information for interpreting and validating threading results. A reliable prediction often needs a great deal of human knowledge and expertise in protein structures and its physicochemical properties. Hence threading programs have often been used in the capacity of assisting human expert predictions.

We have previously developed a procedure for computer-assisted manual prediction of protein fold recognition and structure prediction, based on a threading technique. The effectiveness of this procedure has been extensively tested and refined during the CASP3 and CASP4 prediction seasons. This procedure consists of the following seven main components:

1. **Pre-processing** for identification of protein domains, identification and removal of signal peptides, and protein secondary structure prediction;
2. **Collection of functional/structural information** of a prediction target through various database searches;
3. **Protein triage** for classification of target proteins into membrane proteins, soluble proteins with or without close structural homologues;
4. **Protein fold recognition** for identification of native-like folds and generation of sequence-structure alignments, using threading technique and sequence-based approaches;
5. **Protein structure prediction** for generation of detailed atomic structure models, based on threading alignments;
6. **Structure quality assessment** to evaluate the packing and backbone conformations, and the stereochemical quality of a predicted structure;
7. **Prediction result validation** through comparing predicted structures and collected structural and functional information for consistency check; an iterative prediction/refinement process will be invoked if a significant inconsistency or poor structural quality is detected.

The key goal of this prediction pipeline project is to automate this prediction process, which involves a number of different computational prediction tools, and a process of data interpretation and logic inference by a human predictor.

2.1. Architecture of the Pipeline

Figure 1 illustrates the overall design of our structure prediction pipeline. The **web interface** provides an interactive environment for a user to specify various parameters used in different tools if he/she chooses not to use the default parameter values; it also allows the user to monitor and inject information into the prediction process, and visualize prediction results. The **management system** (or simply **Pipeline Manager**) takes the user-specified parameters and relevant information to start and direct the prediction process; it may select different prediction pathways for each individual protein, based on the available information. After the prediction

pathway is determined, it calls individual prediction tools to execute the prediction process. The final prediction result for each protein target could be a set of predicted structures with associated reliability scores, or an indication that the pipeline did not produce any reliable structure predictions. All results will be displayed through the web interface.

A typical prediction process could be as follows. When a protein sequence is submitted to the pipeline, it will first call prediction tools to identify possible structural domains in the sequence, to identify and remove any signal peptides and to make secondary structure prediction of the target protein, in the **pre-processing** step. It will also invoke its sequence-based search program against various protein databases in the **information collection** step, to collect functional and structural information about the target protein, which will later be used to validate the predicted structures. Then it will predict, in the **protein triage** step, if the protein is a membrane protein or a soluble protein with or without a close structural homologue in PDB (Westbrook et al., 2000). If it is a membrane protein, the prediction process will stop there (a component will be added to make structure predictions for membrane proteins in the near future). If it identifies a close structural homologue in PDB, the pipeline will send the sequence along with its alignment with the homologue to a homology-based modeling program to generate atomic structure models of the protein, in the **homology modeling** step. Otherwise, it will perform protein fold-recognition using information generated by a threading program and sequence-based remote homology detection program, in the **fold recognition** step. The sequence-structure alignments generated in this step will be fed into the **homology modeling** step to generate a detailed 3D model. Then the pipeline will assess the packing and stereochemical quality of the predicted structure model, in the **quality assessment** step. The predicted structure, along with its prediction reliability score and its quality assessment, will be checked against the information already collected, in the **information fusion** step. An iterative refinement process will be invoked if poor structural quality or major inconsistency is detected, by going back to the **fold recognition** step to either re-select the structural folds or re-do the sequence-structure alignment. This process may go through several iterations till some criteria are met.

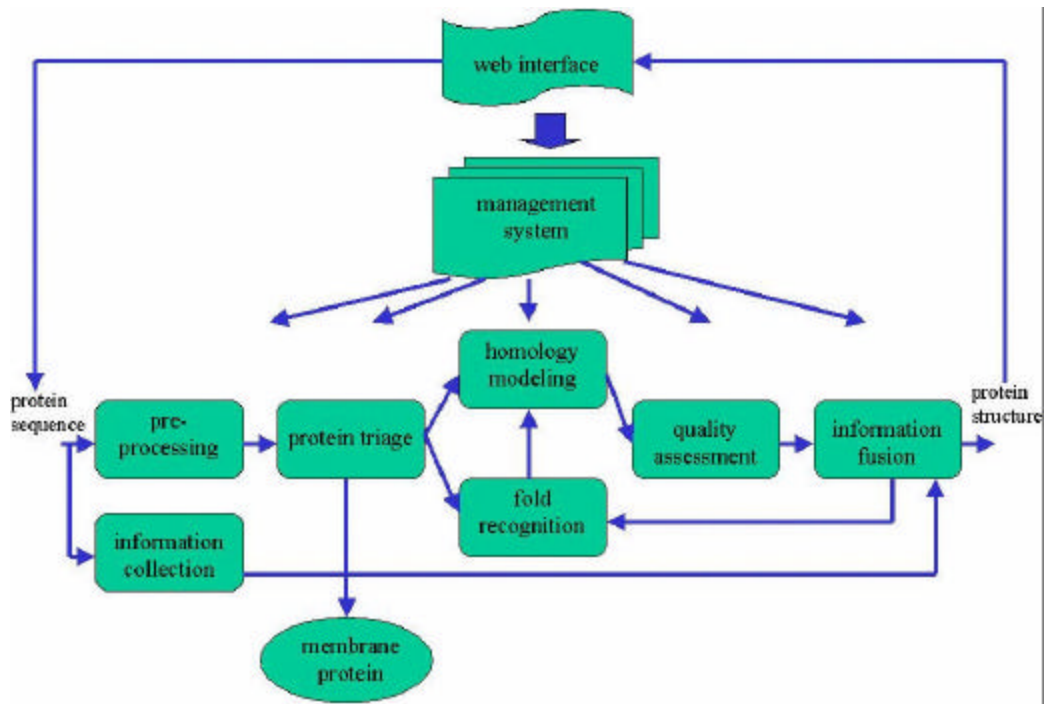


Figure 1: A schematic of the pipeline system for protein structure prediction. Arrows indicate the direction of the prediction process.

2.2. Management System for Prediction Process Control

The management system (Pipeline Manager) encodes some key expert knowledge of human predictors on our CASP3 and CASP4 prediction teams (Xu et al., 1999 and Xu et al., 2001). This is used to guide the prediction process of our automated prediction pipeline. Currently, the Pipeline Manager has the following functionality:

- 1. Decision-making under different conditions.** There are numerous places in the prediction process, where decisions need to be made regarding what to do next and what tools to use. For example, in some cases, a protein may have small transmembrane segments while its major portions are exposed to solvent. In such cases, the system will first label the transmembrane segments and predict their secondary structures; and then remove these segments and make three-dimensional structure predictions on the soluble portions. Such knowledge is encoded as a set of rules that can be applied by the Pipeline Manager.
- 2. Prediction/validation using consensus information.** It has been our experience that if different (independent) prediction tools provide identical or fairly similar structure predictions, this suggests a strong likelihood that the prediction is correct. This is generally true for both the final structure predictions and the intermediate prediction results. For example, our threading program predicts a particular segment of the target protein being an α -helix, based on its alignment with a structural fold. If this segment is also predicted to be an α -helix by a secondary structure prediction program, this should increase our confidence in the secondary structure prediction result. A set of rules, encoding such knowledge, has been developed and is at the disposal of the Manager.
- 3. Integration of functional information.** Proteins having similar functions tend to have similar structures, and this information can be used to validate structure predictions. Functional similarities of proteins are measured based on their relationship (e.g., distance) in the hierarchy of some functional classification scheme, e.g., the Pfam classification (Bateman et al., 2002) or the Enzyme Classification (EC) (Bairoch, 1993). If a protein is identified to belong to a particular EC class and one of its top, say 20, structure predictions is also of the same EC class, we will adjust the rank of this structure prediction to, say top 1, to be consistent with the functional information. A set of rules for integrating functional information into the selection process of structure predictions have been developed and incorporated into the Manager.
- 4. Human inspection and intervention.** Though all structure predictions can be made automatically in the pipeline, our pipeline provides an interface allowing easy human inspection and intervention for possible further improvements. For example, for each target, the pipeline generates a comparison between the secondary structure prediction and the secondary structure assignment from threading alignment. Comparison results between the predicted and assigned secondary structures will allow a human predictor to assess the quality of the prediction and inject needed adjustments in the alignment into the pipeline.

In addition, the Manager is also responsible for communication with the implementation layer (see Section 3) of the pipeline system for service requests (e.g., execution of a particular tool) and for management of the data flow. For example, the Manager will submit a service request for each analysis tool up to three times (in case of service request failures) before it declares a “failure” (see Figure 4) of that tool.

2.3. Tool Selection and Key Features

The following prediction and analysis tools have been selected and deployed to accomplish the designed functionality of different components of the pipeline. Each of these tools has a set of default parameters, suggested by the developers of these tools, which are used as the default values in our pipeline. Our **web interface** allows the user to select different sets of parameters for special purposes, which we will not address in detail in this paper.

A. Pre-processing step

SignalP (Nielsen et al., 1997) identifies the signal peptide in the target protein sequence if there is any, and cut off the peptide at the identified cleavage site. Our pipeline accesses the SignalP program through the Internet (see Section 3 for details).

PRODOM (Corpet et al., 2000) identifies structural domains in a target protein sequence, by searching the known protein domains in the PRODOM database. If database hits were found, it selects the domains with the highest scores among all overlapping domains. As with SignalP, the pipeline accesses PRODOM through the Internet.

SSP (unpublished result, 2002) is an in-house tool for protein secondary structure prediction. It uses a neural network technique to make secondary structure prediction, and its prediction accuracy is comparable to PSI-PRED (Jones, 1999).

B. Information collection step

PSI-BLAST (Altschul et al., 1997) is used to search for relevant functional and structural information of a target protein against various databases, including protein structure database PDB (Westbrook et al., 2000), protein sequence database *nr* (<http://www.ncbi.nlm.nih.gov/BLAST>), ENZYME database (Bairoch, 1993), Pfam (Bateman et al., 2002), etc. Running PSI-BLAST serves two purposes: (a) it can identify homologous proteins to the target protein, from which functional and possibly structural information can be derived; and (b) it can produce reliable sequence alignments with structural homologues if there is any, which can be used for fold recognition and structure prediction in a later stage. Derived functional information will be put into an information pool, which will be used for prediction validation later. We use a local copy of the PSI-BLAST executable code in the pipeline.

C. Protein triage step

The **protein triage** step currently employs two programs, SOSUI and PSI-BLAST, to triage target proteins into three classes: (a) membrane proteins, (b) soluble proteins with close structural homologues, and (c) soluble proteins without close structural homologues.

SOSUI (Hirokawa et al., 1998) is a computer program for identification of transmembrane regions in a protein sequence. Since membrane and soluble proteins have significantly different physiochemical properties, they need different types of prediction programs for their structures. While comparative approaches, like protein threading (see the following), are highly effective for structure predictions of soluble protein, they currently do not apply to membrane proteins. The main reason is that there is only very small number of membrane proteins having solved structures that can be used as structural templates. For each identified transmembrane region (with at least seven residues), the pipeline will remove it and keep the soluble portions, if there are any. Then it treats the soluble portion like a soluble protein. The pipeline accesses SOSUI through the Internet.

The pipeline uses PSI-BLAST to determine if a target protein has a close structural homologue, based on the E-value of the best sequence alignment against the PDB database. A pre-selected E-value threshold (10^{-4}) is used as the default value for such a determination.

D. Protein fold recognition step

The fold recognition component of the pipeline uses information derived from both protein threading method and sequence-based method like PSI-BLAST. It combines these two pieces of information to achieve the optimal prediction result.

PROSPECT (Xu and Xu, 2000) is a protein threading program we have previously developed. PROSPECT has a number of unique features, compared to similar programs. These include that PROSPECT has a unique capability to rigorously deal with residue-residue contact potential, a key energy term for protein fold recognition (Xu and Xu, 2000). Also PROSPECT has a unique way of incorporating evolutionary information into its all energy terms, including singleton energy term, residue-residue contact term and mutation energy term (Kim et al., 2002). These unique features have made the program perform significantly better, in fold recognition, than any other existing threading programs. The following figure shows a comparison of fold recognition performance among different programs (Kim et al., 2002).

Method	%pairs at top 1/top 5		Family only		Superfamily only		Fold only	
	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5
PROSPECT	84.1	88.2	52.6	64.8	27.7	50.3		
FUGUE	82.2	85.8	41.9	53.2	12.5	26.8		
PSI-BLAST	71.2	72.3	27.4	27.9	4.0	4.7		
HMMER-PSIBLAST	67.7	73.5	20.7	31.3	4.4	14.6		
SAMT98-PSIBLAST	70.1	75.4	28.3	38.9	3.4	18.7		
BLASTLINK	74.6	78.9	29.3	40.6	6.9	16.5		
SSEARCH	68.6	75.7	20.7	32.5	5.6	15.6		
THREADER	49.2	58.9	10.8	24.7	14.6	37.7		

Figure 2: A comparison between PROSPECT and other fold recognition programs on a large benchmark set (Kim et al., 2002), at different sequence similarity levels: family, superfamily and fold family. The benchmark set consists of 600 proteins and the template set consists of ~3000 proteins. “Top 1” means that the percentage of the correct folds is placed as the highest ranked folds, and “Top 5” means that the percentage of the correct folds are placed among the five highest ranked folds for each target protein.

Additional unique features of PROSPECT include a capability for a user to incorporate various types of protein-specific information into the threading process as constraints. These may include (a) predicted or known secondary structure information, (b) residue-residue distance information, possibly derived from known disulfide bonds or other experiments like NMR (Xu et al., 2001). For each fold prediction, PROSPECT provides a zscore that measures the reliability of the prediction. The zscore is calculated using a support vector machine (SVM) approach, which combines various threading energy scores, features of the target protein sequence and the template structure, including the sequence lengths and the compactness of the structure, and the

E-value of PSI-BLAST alignment. We refer the reader to (Kim et al., 2002) for detailed discussion of this work.

E. Homology modeling step

Two programs are used to generate detailed atomic structures, using sequence-structure alignments generated either by PROSPCT or PSI-BLAST (for proteins with close homologues). A user can select the option of using one or both of the programs.

MODELLER (Sali and Blundell, 1993) is a popular homology modeling program. It generates both backbone structure and side-chain packing conformation based on provided sequence-structure alignments. In case of missing loop regions in the provided alignments (which may often be the case when the target protein does not have a close structural homologue), MODELLER can add its predicted loop structure. A user can specify the number of structure models that are desired to be generated with the lowest potential energy.

NEST/JACKAL (Xiang and Honig, 2001) is a new software for detailed atomic structure prediction using provided sequence-structure alignments, similar to MODELLER. Our experience has been that its performance is comparable to that of MODELLER but runs faster. One key advantage of this software is that it is a freeware.

Currently, homology modeling is the slowest step in the whole pipeline. On average, it takes about 80% of the prediction time for each protein.

F. Quality assessment step

WHATIF (Vried, 1990) provides a capability to evaluate the structural quality of a predicted structure. This includes the quality of side-chain packing and backbone conformations, the inside/outside occupancies of hydrophobic and hydrophilic residues, and stereochemical quality of the predicted structure. Two quantities are calculated to measure the overall quality of a structure: (P1) the sum of the 2nd generation packing quality and the quality of the Ramachandran plot, and (P2) the quality of backbone conformation. For any two models X and Y, we use the following rule to rank their qualities: X is considered of better quality than Y if $P1(X) > P1(Y)$, or if $P1(X) = P1(Y)$ and $P2(X) > P2(Y)$.

G. Information fusion step

This step consists of a set of rules for (a) cross-validating predictions of native-like folds and protein structures using information derived from different sources, and (b) ranking and selecting the final fold and structure predictions. This set of rules has been implemented in an in-house program, called **HitEvaluator**. We now list some of the rules to illustrate the basic idea:

A structure template will be selected as a recognized structural fold if

- 1) the E-value of the PSI-BLAST alignment between the template and the target protein is < 0.02 , or
- 2) the E-value is < 1.0 and its raw threading score or the z-score is ranked among the top 50 structural templates, or
- 3) the template's EC number matches at least the first three digits of the EC number of the target protein and the raw threading score or the z-score of the template is ranked among the top 50 templates, or

- 4) the ranks of its raw threading score and the z-score are both among the top 20 predicted folds.

After a template structure has been selected, we assign a reliability score as follows:

1. if the E-value of the PSI-BLAST alignment between the template and the target protein is < 0.02 , add the reliability score by 10, or
2. if the EC number of the template matches every digit of the EC number of target protein, add the reliability score by 5, or
3. add 2 point to the reliability score for every keyword match between the functional annotations of the template and the target protein.

All rules are derived based on our prediction experience gained through CASP predictions and other prediction applications (Xu et al., 2000; Xu et al. 2002). The **information fusion** step also identifies the portions of a predicted structure with poor Ramachandran plot, which will be used to guide an iterative process for improvement of the threading alignment.

2.4. Web Interface Design

The web interface provides an interactive environment and capability for a user to

1. input the target sequence and related information;
2. specify prediction tools to be used and define the prediction pathway (which can overrule the prediction pathway generated by the Pipeline Manager);
3. modify the default parameter values of selected prediction and analysis programs;
4. monitor the progress and status of the prediction pipeline;
5. inject protein-specific information into the prediction process, interactively; and
6. visualize and examine the computational results of each tool.

3. SYSTEM DESIGN AND IMPLEMENTATION

This section discusses how the prediction pipeline, consisting of multiple tools running on different local and remote computers, is implemented as a system. One key design principle is to shield the end-user from the complexities of the use of a diverse set of tools, hardware and software requirements of each tool, the need to remember the syntax for invoking individual tools, operation in a heterogeneous computing environment, and other issues.

The pipeline architecture has been designed with the aim of providing the central decision-making application with the framework of modular objects representing applications distributed over the local network and Internet. All the tool interoperability issues are hidden behind module interfaces, which makes it effortless to incorporate new tools. The server handles all the complexities of request validation, queuing of multiple, concurrent requests, load balancing, running each request on an appropriate compute server from the server pool available to the system. Major parts of the system are written in Perl. The pipeline system is realized through three layers of implementation. (a) the protein analysis toolkit (PAT), (b) the client server layer for remote access to PAT, and (c) the integrated supercomputing toolkit to distribute computational jobs to a suite of heterogeneous computing platforms. Figure 3 shows a schematic of the system architecture of the pipeline.

The entire prediction process is coordinated by the Pipeline Manager. The Pipeline Manager invokes different tools based on the user input and the logic of the prediction process, and

controls the data and analysis flow of the pipeline. Each tool is represented as two modules, one being the service responsible for data transfer and data-format conversion between pipeline and particular application, and another being the tool itself wrapped into the service that dispatches the jobs between available servers. All such modules are independent of each other so a user can choose to run just the tools he/she wants to. The Manager triggers execution of the tool, and corresponding modules communicate with the central repository of the pipeline results, which is implemented as a single XML file. XML (eXtensible Markup Language) has been extensively used in the system to facilitate standardized data formats, robust data validation and ease of data exchange between GUI, Pipeline Manager and all services.

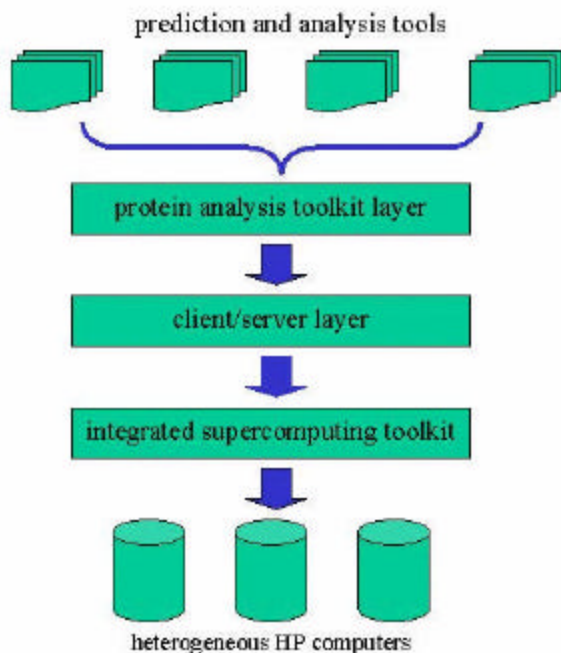


Figure 3: A schematic of the system architecture of the structure prediction pipeline.

3.1. Protein analysis toolkit (PAT)

The suite of prediction and analysis tools currently used by the prediction pipeline (and a number of tools that we plan to add in the near future) use different programming languages, have different execution resource requirements, need different supported hardware and operating system platforms, and have different input, output and parameter formats. Additionally, individual tools may be available as binary code, source code, or may be accessible only via a web CGI interface. We have developed the PAT Toolkit to abstract out this diversity by providing a generic, consistent service wrapper interface to all the tools incorporated in the toolkit.

All tools are compiled for the supported computer architectures and operating systems. Highly compute intensive ones, like PROSPECT, have been ported on ORNL supercomputers. Some tools developed by other labs are accessible only via web servers. Web agents have been implemented to access these tools. Access to the system, at the most basic level, is via a single, generic Perl client script, which can run on any platform that has a Perl installation. The only requirement is that each analysis request be packaged in a pre-specified XML format, for a given

analysis service, conforming to the service description. The public interface to the system is in the form of service description for each tool incorporated in the Toolkit.

XML has been extensively used in the implementation of the toolkit and the client server system. Service specifications are written in XML; all service requests are required to be in XML; and the service results are encapsulated as XML documents. XML service specification for each tool consists of service name and version, required and optional inputs and required or optional parameters, along with valid parameter values or range of values, as appropriate.

Each tool is accessed via a corresponding Perl service wrapper script, which facilitates several important functions. The service wrapper script determines the operating system it is invoked in and executes the appropriate architecture-specific tool executable code. It also validates the input request based on the corresponding XML service specification. Additionally, it performs input data and parameter transformations, as necessary, and on completion of tool execution, it packages the results into a pre-specified XML output format.

3.2. Client Server System for Remote Access to PAT

The client-server system facilitates efficient handling of a large number of prediction and analysis tasks submitted to the system. The client-server protocol incorporates issuance of a ticket (request ID), which eliminates the need for maintaining persistent client-server connections, allowing the client to retrieve the results later, at the same time, conserving server resources. The server distributes the incoming requests intelligently on the available heterogeneous pool of compute server machines, based on the loads on the various servers. Highly compute-intensive service requests (like PSI-BLAST and PROSPECT) are transparently redirected to the high-performance server running on ORNL's IBM RS/6000 SP supercomputer and ORNL's TORC Linux cluster (with 64 nodes) through GIST.

3.3. The Genomic Integrated Supercomputing Toolkit (GIST)

GIST is a framework for large-scale biological application deployment, used to provide a transparent, distributed and fault tolerant interface to biological applications and data sets, upon the wide diversity of supercomputing platforms. In the context of interactive online interfaces to biological applications, it provides a single point of entry to utilize a scalable multi-site high-performance computing enterprise, but also highly fault tolerant behavior. The PAT layer sends requests to GIST for execution of prediction/analysis tools on ORNL supercomputers.

The system components consist of a single entry point named Chameleon, which is a component that uses an XML like template description of application behavior to emulate interaction with the real application, remotely. Hence, all tools retain and in some cases have enhanced behavior, via the command line or even API reference. Since requests can be validated up front, this is an extremely portable and transparent method of using supercomputing resources. The Chameleon tool, thus communicates with remote servers by means of an opaque object structure, which is deconvoluted at the remote side, to reproduce exactly an instance of the requested application. The advantage of having a self-consistent object is that it is able to run on any system that supports derived dependencies, calculated by Chameleon.

There are 2 basic modes with which we have been able to implement a system suitable for both large-scale batch operations, as well as maintaining an interactive environment suitable for command line and WWW usage.

- A. **The tier algorithm:** This presents the resources as a selection of tiers, which can be mapped to hardware resources, applications or even individual projects. The most important feature is that requests can “trickle down” onto lower tiers upon the exclusion, overload or even location-based preferences of the best place to execute a transaction. This configuration performs most effectively for maximum fault tolerance.
- B. **The pool algorithm:** This presents the resources as a single pool, which can of course be composed of tiers if required. The advantage of this configuration is enhanced for scalability where many applications of widely differing average response times, can share the same logical space. Since the slowest portion of a pipeline, limits its overall throughput, replicated resources can be assigned to reduce the time to solution by effectively making the slowest portion comparable to other portions. This is of particular concern with such calculations involving threading such as PROSPECT, as the solution time for many are unpredictable.

3.4. Web interface

The structure prediction pipeline currently consists of two types of interfaces: (a) an interactive web interface, and (b) an interface of fully automated pipeline computation.

A. Interactive web interface

Web interface is implemented as a number of dynamic pages generated by the CGI processes. CGI is also used to trigger the job, generate unique ID for it to follow the execution status of the process. At any given moment it provides access to the up-to-date pipeline results. The interface allows the user to configure the pipeline by selection of individual tools and tool-specific parameters, and loading the input sequence. Upon submission of the request, the interface provides continuous feedback to the user about the pipeline execution progress status, by periodic update of the status web page. The status web page also allows the user to inspect the results of the individual tools, during and after the execution of the pipeline request.

The pipeline interface form is dynamically generated by on the fly lookup of the pipeline and individual service specification (XML) documents. This functionality ensures that any updates to the service specifications or addition of new services are automatically reflected in the web interface, without having to perform any manual modifications.

B. Interface for automated pipeline

Typically the process of protein analysis is triggered from the Web interface, although in case of batch processing command line interface could be used.

Web browser client is used to send query sequence to the server, choose parameters and the path for the analysis, trigger the execution, and browse status of the process and results. User can run all the available tools or select just few of them, choose parameters of the protein in question as well as the parameters used by each individual pipeline tool.

Pipeline assigns unique Query ID to each job submitted to the server. This ID could be used to retrieve execution results after browser had been closed. All the data related to the particular Query ID is kept at the server for extended period of time.

The GUI provides access to the number of dynamically generated pages containing information about parameters used by the tools, status of the job processing and error messages issued by the system. All the tools are applied to the target sequence or to its subsequences, and status page

reflects the status of analysis of each domain (see Figure 4). It indicates if the domain is still processed by a particular tool, or how successful the analysis was. This page also has links to the pages with more detailed information about each application's processing status. After all domains had been processed, system considers the whole pipeline run successful if there was no failure of any tool on any sequence segment.

The status page is linked to all the results of the tool execution. Page of the tool results displays prediction made by the tool and explains decisions made by Pipeline Manager based on these predictions. As a rule, first page displays information essential for the decision-making process and structure prediction, but it also provides links to the rest of the results. In order to make results analysis more convenient user may interact with GUI to choose the way data is presented. Pipeline extracts in advance useful information from the various databases and supplies results of some applications with this data. For instance, template enzyme classification number and function annotation are added to the Prospect threading results.

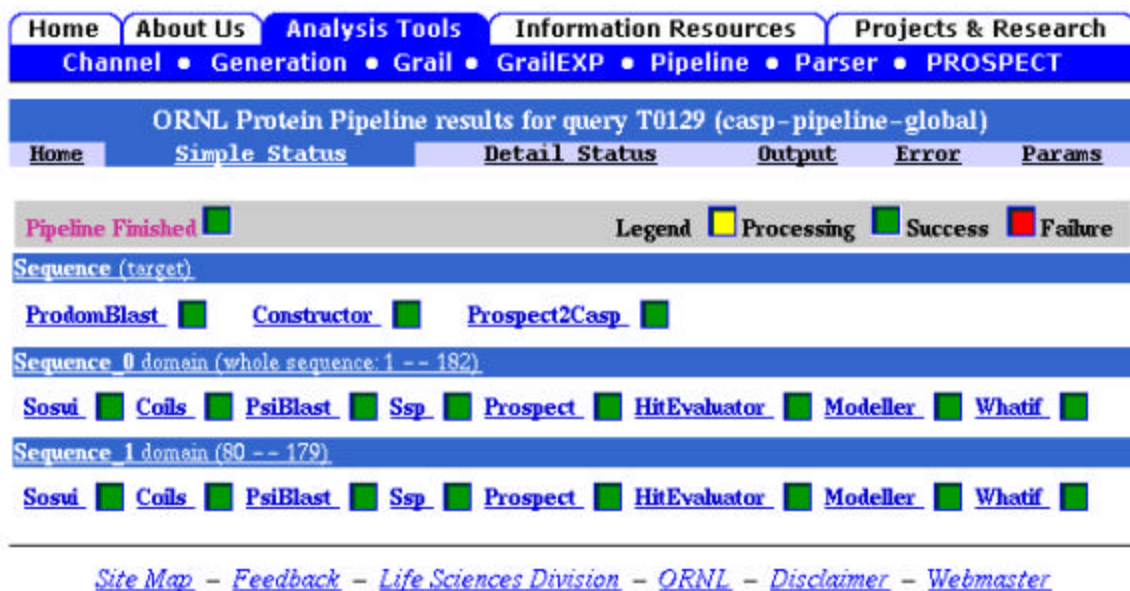


Figure 4: A screen shot of the web interface of the structure prediction pipeline.

4. APPLICATIONS OF THE PIPELINE

A number of large-scale applications, using the structure prediction pipeline, have been carried out. We now outline two such applications.

4.1. Genome-scale prediction on *Caenorhabditis elegans* and *Pyrococcus furiosus*

In collaboration with SouthEast Collaboratory for Structural Genomics (SECSG), we predicted structures for ~1300 proteins in *Caenorhabditis elegans* and its microbial ancestor, *Pyrococcus furiosus*. These proteins do not have any significant BLAST hits in PDB as run by researchers at SECSG. The goal of this application is to determine if any of them may have structural homologues in PDB so will be removed from their target list for experimental solution of protein structures. By running these proteins through our structure prediction pipeline, we have found that

- 1) 58 out of the 1300 target proteins are membrane proteins as predicted by SOSUI;
- 2) 134 (10%) of the soluble proteins have PSI-BLAST hits against PDB with E-values < 0.0001 , indicating that they have structural homologues;
- 3) additional 135 (10%) soluble proteins have PROSPECT hits against PDB with z-score > 20 , indicating that the confidence level for these fold recognition is $> 99\%$ (Figure 5 shows the prediction specificity and sensitivity versus z-score of PROSPECT);
- 4) another 60 (5%) soluble proteins have PROSPECT hits against PDB with z-score > 12 but < 20 , indicating that the confidence level for these fold recognition is between 96% to 99%; and
- 5) another 76 (6%) of soluble proteins have PROSPECT z-scores > 8 and < 12 , indicating that these fold recognition have at least 63% of chance to be correct.

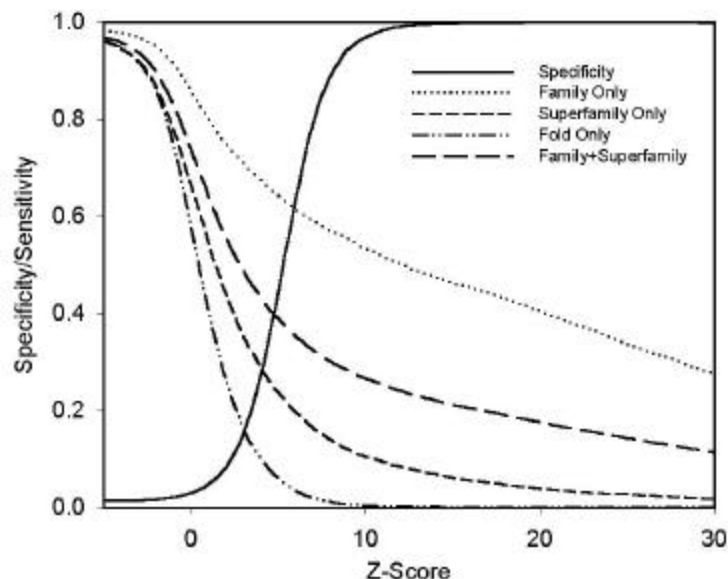


Figure 5: Z-score versus prediction specificity and sensitivity (Kim et al., 2002). The solid line represents the prediction specificity. For example, a z-score of 10 indicates that the probability for the prediction to be correct is about 0.95.

Overall, the prediction pipeline predicted 31% of the 1300 proteins to have structural homologues in PDB, none of which have apparent BLAST hits in PDB. This is consistent with our previous observations that our prediction pipeline can reliably identify structural homologues for about one third of all proteins that BLAST did not find any structural homologues. Figure 6 shows the distribution of computing time (in minutes) on a single Linux processor versus the lengths of 1300 proteins.

4.2. Structure prediction in CAFASP and CASP

We have used the prediction pipeline for automated protein structure prediction in the third CAFASP (Critical Assessment of Fully Automated Structure Prediction, <http://www.cs.bgu.ac.il/~dfischer/CAFASP3/>) as part of the CASP5 structure prediction contest (<http://predictioncenter.llnl.gov/casp5/>). Our pipeline made structure prediction for all 67 prediction-targets. Though the evaluation results on CASP predictions have not been announced yet, the following example provide some indication about the effectiveness of combining multiple sources of information to make a prediction by our prediction pipeline.

Target t0136 is the transcarboxylase 12S subunit with 523 residues and its EC number is EC: 4.1.1.41, based on our search result. The initial prediction by our prediction pipeline is that it has the same structural fold of 1a53 (PDB code), which has the EC number of EC:4.1.1.48, indicating that the two proteins have similar functions. However the threading scores between 1a53 and t0136 are not particularly high due to the poor threading alignment between the two proteins. Then through 1a53, we found that another protein 1thfd of the same structural fold of 1a53 (as indicated by their FSSP numbers), which gives better threading alignment with t0136. This protein was not originally identified because it does not have a known EC number, and hence no functional information can be used directly in the initial fold recognition process. 1thfd was submitted as the top candidate of t0136.

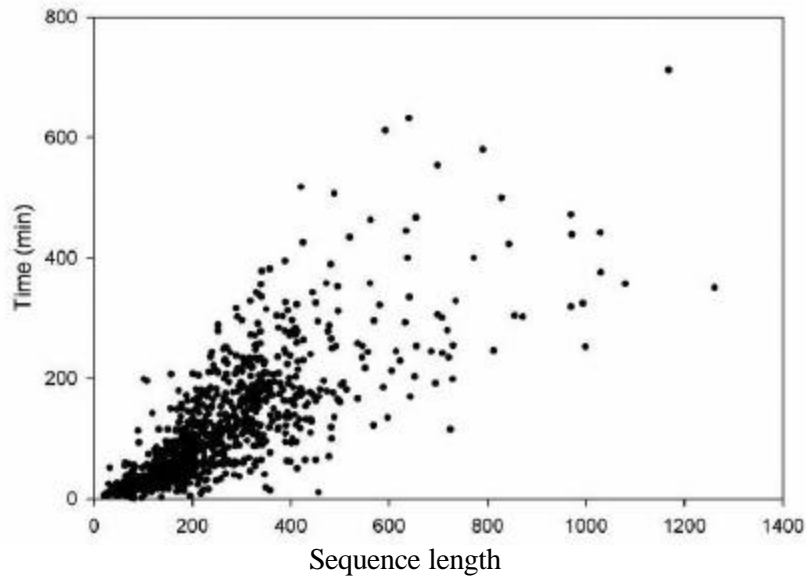


Figure 6: Computational time (in minutes) of structure prediction by the pipeline versus protein sequence length.

5. SUMMARY

We have developed an automated pipeline for protein structure predictions. The pipeline has been implemented on both the supercomputers and a 64-node Linux cluster at ORNL. On the 64-node Linux cluster, the pipeline is capable of making genome-scale structure predictions for a microbial genome with 2,000 – 5,000 genes within a week or two. Typically, ~60% of the genes in a microbial genome can have some level of reliable structural prediction: structural fold recognition, backbone structure prediction, or detailed atomic structure prediction with side-chains, while the reliability of each prediction is indicated by the z-score of the prediction. The pipeline is currently available for general public service, which can be accessed at <http://compbio.ornl.gov/proteinpipeline/>.

ACKNOWLEDGEMENTS

This research was sponsored by the Office of Health and Environmental Research, U.S. Department of Energy, under Contract No. DE-AC05-000R22725 managed by UT-Battelle, LLC. We thank Dr. B.C. Wang and Dr. Dawei Lin of University of Georgia at Athens for sending us

the 1300 protein sequences for structure predictions. The authors also thank Dr. Al Geist of ORNL for providing us the full access to the XTORC Linux cluster for the protein structure prediction pipeline.

REFERENCES:

1. Alexandrov NN, Nussinov R, Zimmer RM. Fast protein fold recognition via sequence to structure alignment and contact capacity potentials. *Pac Symp Biocomput.* 1996:53-72.
2. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ, Basic local alignment search tool, *J Mol Biol*, 1990, 215(3):403-10.
3. Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., Lipman, D. J. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25:3389-3402.
4. Bairoch A, The ENZYME data bank, *Nucleic Acids Research*, 1993, 21:3155-3156.
5. Bateman A, Birney E, Cerruti L, Durbin R, Etwiller L, Eddy SR, Griffiths-Jones S., Howe KL, Marshall M, and Sonnhammer ELL, *Nucleic Acids Research*, 2002, 30(1):276-280.
6. Bowie JU, Luthy R, and Eisenberg D. A method to identify protein sequences that fold into a known three-dimensional structure. *Science*, 1991, 253:164--170.
7. CASP, Protein Structure Prediction Issue. *Proteins: Struct. Funct. Genet.*, 1995. 23:295--462.
8. CASP, Protein Structure Prediction Issue, *Proteins: Struct. Funct. Genet.*, 1997, Suppl. 1. 29:1—230.
9. CASP, Protein Structure Prediction Issue. *Proteins: Struct. Funct. Genet.* 1999, Suppl. 3. 37:1-237.
10. CASP, Protein Structure Prediction Issue. *Proteins: Struct. Funct. Genet.* 2001, Suppl. 4.
11. Chance MR, Bresnick AR, Burley SK, Jiang JS, Lima CD, Sali A, Almo SC, Bonanno JB, Buglino JA, Boulton S, Chen H, Eswar N, He G, Huang R, Ilyin V, McMahan L, Pieper U, Ray S, Vidal M, Wang LK. Structural genomics: a pipeline for providing structures for the biologist. *Protein Sci.* 2002 Apr;11(4):723-38.
12. Corpet F, Servant F, Gouzy J, and Kahn D, ProDom and ProDom-CG: tools for protein domain analysis and whole genome comparisons, *Nucleic Acids Res.* 2000, 28:267-269.
13. Hirokawa T, Boon-Cheing S, and Mitaku s, Classification and secondary structure prediction system for membrane proteins, *informatics*, 1998, -379.
14. Jones DT, Taylor WR, and Thornton JM. A new approach to protein fold recognition. *Nature*, 1992, 358:86-89.
15. Jones DT, Protein secondary structure prediction based on position-specific scoring matrices, *Mol. Biol.*, 1999, 5-202.
16. Kim D, Xu D, Guo J, Ellrott K, Xu Y, PROSPECT II: protein structure prediction program for genome-scale application”, submitted, 2002.
17. Kitson DH, Badretdinov A, Zhu ZY, Velikanov M, Edwards DJ, Olszewski K, Szalma S, Yan L. Functional annotation of proteomic sequences based on consensus of sequence and structural analysis. *Brief Bioinform.* 2002 Mar;3(1):32-44.
18. Krogh A, Larsson B, von Heijne G, Sonnhammer EL. Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. *J Mol Biol.* 2001 Jan 19;305(3):567-80.
19. Lander EC, et al. Initial sequencing and analysis of the human genome, *Nature*, 2001, 409:860-921.
20. Lupas A., Van Dyke M., Stock J., Predicting Coiled Coils from Protein Sequences, *Science*, 1991;252:1162-1164.

21. Montelione GT and Anderson S. Structural genomics: keynote for a Human Proteome Project. *Nature Struct. Biol.*, 1999, 6:11 -- 12.
22. Murzin AG., Brenner SE, Hubbard T, and Chothia C. Scop: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, 1995, 247:536--540.
23. Nielsen, H., Engelbrecht, J., Brunak, S., von Heijne, G. 1997. Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Protein Eng.* 10:1-6.
24. Sali, A. Blundell, T. L. 1993. Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.* 234:779-815.
25. Sanchez R, Sali A. Large-scale protein structure modeling of the *Saccharomyces cerevisiae* genome. *Proc Natl Acad Sci U S A.* 1998 Nov 10;95(23):13597-602.
26. Schaffer AA, Wolf YI, Ponting CP, Koonin EV, Aravind L, Altschul SF. IMPALA: matching a protein sequence against a collection of PSI-BLAST-constructed position-specific score matrices.
27. Sippl MJ and Weitckus S. Detection of native-like models for amino acid sequences of unknown three-dimensional structure in a database of known protein conformations. *Proteins: Struct. Funct. Genet.*, 1992, 13:258--271.
28. Venter VC, et al, The sequence of the human genome, *Science*, 2001, 291:1304-51.
29. Vriend G, HATIF: a molecular modelling and drug design program, *J. Mol. Graphics*, 1990, 8:52-56.
30. Westbrook J, et al, The Protein Data Bank: unifying the archive, *Nucleic Acids Res*, 2000, 30(1):245-248.
31. Xiang Z, Honig B, Extending the accuracy limit of side-chain prediction, *J. Mol. Biol.* 2001, 311:421-430
32. Xu Y and Xu D, Protein Threading using PROSPECT: design and evaluation, *Protein: Structure, Function, Genetics*, 2000, Vol 40, pp 343 - 354.
33. Xu D and Xu Y. Computational Studies of Protein Structure and Function Using Threading Program PROSPECT. In *Protein Structure Prediction: Bioinformatic Approach*, edited by Igor Tsigelny. International University Line publishers (IUL), 2002, La Jolla, CA. Pages 5-41.
34. Xu Y, Xu D, Crawford O, and Einstein JR, A computational method for NMR-constrained protein threading, *Journal of Computational Biology*, 2000, Vol 7(3/4), 449 – 467.
35. Xu D, Crawford OH, LoCascio PF, and Xu Y. Application of PROSPECT in CASP4: Characterizing Protein Structures with New Folds. *Proteins: Structure, Function, and Genetics (CASP4 Special Issue)*. 2001, 46:140-148.
36. Xu D, Baburaj K, Peterson CB, and Xu Y, A Model for the Three Dimensional Structure of Vitronectin: Predictions for the Multi-Domain Protein from Threading and Docking, *Proteins: Structure, Function, Genetics*, 2001, vol. 44: 312-320.
37. Xu Y, Xu D, Crawford OH, Einstein JR, Larimer F, Uberbacher EC, Unseren MA, and Zhang G. Protein threading by PROSPECT: a prediction experiment in CASP3. *Protein Engineering*, 1999. 12:899-907.