Computational Tools For Protein Modeling

Dong Xu* , Ying Xu and Edward C. Uberbacher

Computational Biosciences Section Life Sciences Division Oak Ridge National Laboratory Oak Ridge, TN 37831-6480, USA

Abstract: Protein modeling is playing a more and more important role in protein and peptide sciences due to improvements in modeling methods, advances in computer technology, and the huge amount of biological data becoming available. Modeling tools can often predict the structure and shed some light on the function and its underlying mechanism. They can also provide insight to design experiments and suggest possible leads for drug design. This review attempts to provide a comprehensive introduction to major computer programs, especially on-line servers, for protein modeling. The review covers the following aspects: (1) protein sequence comparison, including sequence alignment/search, sequence-based protein family classification, domain parsing, and phylogenetic classification; (2) sequence annotation, including annotation/prediction of hydrophobic profiles, transmembrane regions, active sites, signaling sites, and secondary structures; (3) protein structure analysis, including visualization, geometry analysis, structure comparison/classification, dynamics, and electrostatics; (4) three-dimensional structure prediction, including homology modeling, fold recognition using threading, *ab initio* prediction, and docking. We will address what a user can expect from the computer tools in terms of their strengths and limitations. We will also discuss the major challenges and the future trends in the field. A collection of the links of tools can be found at http://compbio.ornl.gov/structure/resource/.

1 INTRODUCTION

Computational tools for protein modeling are playing a more and more important role in protein and peptide sciences, from the genome scale to the atomic level. As molecular biology is moving toward genome scale, a huge amount of biological data is being generated. Particularly, the Human Genome Project and other genome sequencing efforts are providing DNA sequences at a prodigious rate, and these sequences are yielding tens of thousands of new genes and proteins. Sequence comparison and other analysis using computational tools can identify the function or the structure of a protein by recognizing its relationship to other proteins in the databases. Various prediction programs/servers can annotate function/structure information for many hypothetical proteins. Protein modeling tools can also be used to study biochemical processes, such as enzyme reactions and electron transfer. Although spectroscopy methods can measure these processes, usually the details of the underlying mechanisms cannot be shown directly based on experimental methods alone. Using computer simulations to bridge the gap between experimental data and theoretical models often provides the whole picture. It is widely recognized that protein modeling is an indispensable part of modern molecular biology.

Protein modeling is a very active field. Recognition of its importance has led increased funding for the research and development of protein modeling methods and tools. Many researchers from diverse backgrounds, such as mathematics, physics, chemistry, biology, computer science, and engineering, have entered this inter-disciplinary area. As a result, new developments in recent years have made protein modeling more reliable, efficient, and user-friendly. Meanwhile, computers are becoming substantially faster, and the price of hardware, such CPU, memory, and storage, is plummeting. While cutting-edge computing efforts may tackle large-scale biomolecular modeling using parallel machines or network clusters, small research groups can easily apply modeling tools using affordable computers. In addition, the Internet provides an efficient way to do protein modeling. Protein modeling packages are distributed throughout the

*Address correspondence to this author at Computational Biosciences Section, Oak Ridge National Laboratory, 1060 Commerce Park Drive, Oak Ridge, TN 37830-6480. Email: xud@ornl.gov. Fax: 423-241-1965.

1389-2037/00 $25.00+.00

Internet. The Web servers for proteins allow users worldwide to access up-to-date software and databases, with easily mastered interfaces. To use such servers, researchers do not have to understand the Unix operating system or own a powerful workstation. Many protein servers are becoming popular in protein research. For example, the SignalP server [1], which predicts signal peptides and their cleavage sites from protein sequences, represents one of the most quoted papers in the past few years. As of June, 1999, it had been cited by more than 250 papers [2] since it was published in January, 1997.

This paper reviews the computational tools for different aspects of protein modeling, including the major methods and computer programs in sequence comparison and annotation, as well as structure analysis and prediction. Among hundreds of protein modeling tools, we only select a few widely used ones in each category as illustrative examples. A number of excellent reviews, which are cited in the following sections, have summarized different aspects of protein modeling tools. However, to our knowledge, this review is the first effort to comprehensively overview all types of protein modeling tools. The following sections provide an introduction to (1) what protein modeling tools are available, (2) how they work (methods and algorithms), and (3) what results a user can expect (sensitivity and reliability). We also describe current developments for each type of tool and approaches to combining different types of tools to solve biological problems. The strength, pitfall, and future directions of the major types of protein tools will be addressed. The Web addresses of representative tools are listed in Tables 1-4.

The rest of the review is organized as follows: section 2 introduces tools based on sequence-sequence comparison; section 3 addresses tools that annotate and predict properties for a sequence; section 4 discusses analysis tools for a given structure; section 5 reviews three-dimensional (3D) structure prediction tools. Finally, we summarize the general issues of using protein tools in Section 6.

2 SEQUENCE COMPARISON

Sequence comparison is typically the starting point for analysis of a new protein [3]. Because of the exponential growth in sequence data, sequence comparison becomes a more and more powerful tool. Relating a protein sequence to other sequences often reveals its function, structure, and evolution. However, it should be noted that sequence comparison is based on sequence similarity which may not always correspond to biological relationship (homology), especially when the confidence level of a comparison result is low. Also, homology does not always mean function conservation. In this section, we will discuss pairwise/multiple sequence alignment, sequence family, domain parsing, phylogenetic classification, and sequence search methods.

2.1 Pairwise Sequence Alignment

Pairwise sequence comparison is the major approach to finding possible homologs for a protein in sequence databases such as *SWISS-PROT* [4], *TrEMBL* [4], and *PIR* [5]. It is also the foundation for more complex sequence comparison methods. A pairwise sequence alignment compares two protein sequences according to a match criterion, which is expressed in a 20-by-20 mutation matrix with elements (*i*; *j*), describing the preference (score) to replace the amino acid type *i* with *j*. Several matrices have been developed based on mutation rates found in sequence databases, and the most popular ones are the PAM [6] and BLOSUM [7] matrices. To use which matrix depends on the purpose of the sequence alignment. The BLOSUM-62 is a widely used matrix for searching close homologs. However, for identifying remote homologs, it is probably better to choose PAM250 [8], which represents the transition probabilities between amino acids with 250 accepted mutations per 100 amino acids.

Several types of algorithms are used to obtain the optimal or near-optimal alignment given a mutation matrix with penalties for the insertion/deletion of gaps in the alignment. The first well-known algorithm was developed by Needleman and Wunsch [9], who applied the dynamic programming technique to determine the optimal solution for a global alignment. The method was improved by Smith and Waterman [10] so that similarity between short segments of the two sequences (local alignment) can be identified more efficiently in a way that guarantees to find the optimal solution. It has been implemented in *SSEARCH*, in *SKESTREL* with the specialized hardware design [11], and in the *BESTFIT* module of the *GCG* package [12]. Heuristic search algorithms, e.g., the ones used in the popular programs *FASTA* [13] and BLAST [14], are less sensitive but much faster than the Smith-Waterman algorithm. *FASTA* allows insertion of gaps during the alignment phase (a way that simulates insertions

Table 1. Selected Sequence Comparison Tools

Pairwise Sequence Alignment		
ALIGN	www2.igh.cnrs.fr/bin/align-guess.cgi	server
BLAST	www.ncbi.nlm.nih.gov/BLAST/	server/executable
FASTA	www.embl-heidelberg.de/cgi/fasta-wrapper-free/	server
GCG/BESTFIT	www.gcg.com	executable
KESTREL	www.cse.ucsc.edu/research/kestrel/	server
SSEARCH	vega.igh.cnrs.fr/bin/ssearch-guess.cgi	server
Multiple Sequence Alignment		
BCM Search Launcher	dot.imgen.bcm.tmc.edu:9331/multi-align/	server
BlockMaker	blocks.fhcrc.org/blocks/blockmkr/	server
CLUSTAL	ubik.microbiol.washington.edu/ClustalW/	executable
CypData	ftp.genome.ad.jp/pub/genome/saitama-cc/	executable
GCG/PILEUP	www.gcg.com	executable
MEME	www.sdsc.edu/MEME/meme/website/	server
Multalin	www.toulouse.inra.fr/multalin.html	server
PAUP*	www.lms.si.edu/PAUP/	executable
Sequence Family		
BLOCKS	www.blocks.fhcrc.org	server
COG	www.ncbi.nlm.nih.gov/COG/	server
DOMO	www.infobiogen.fr/services/domo/	server
MEGACLASS	www.ibc.wustl.edu/megaclass/	server
Pfam	www.sanger.ac.uk/Pfam/	server
PRINTS	www.biochem.ucl.ac.uk/bsm/dbbrowser/PRINTS/	server
ProClass	pir.georgetown.edu/gfserver/proclass.html	server
ProDom	protein.toulouse.inra.fr/prodom.html	server
PROSITE	www.expasy.ch/prosite/	server
SBASE	www2.icgeb.trieste.it/~sbasesrv/	server
Phylogenetic Classification		
MOLPHY	dogwood.botany.uga.edu/malmberg/software.html	executable
PAML	abacus.gene.ucl.ac.uk/ziheng/paml.html	executable
PASSML	ng-dec1.gen.cam.ac.uk/hmm/Passml.html	executable
PHYLIP	evolution.genetics.washington.edu/phylip.html	executable
PUZZLE	members.tripod.de/korbi/puzzle/	executable
TAAR	www.dcss.mcmaster.ca/~fliu/taar download.html	executable
TOPAL	www.bioss.sari.ac.uk/~grainne/topal.html	executable
Search Based on Multiple Sequence Alignment		
HMMER	hmmer.wustl.edu	executable
PSI-BLAST	www.ncbi.nlm.nih.gov/BLAST/server/	executable
SAM-T98	www.cse.ucsc.edu/research/compbio/HMM-apps/	server

and deletions during evolutionary divergence) to maximize the number of aligned residues. It works well for global alignment. *BLAST* is the most widely used local alignment tool. It is also the fastest tool generally available (a pairwise alignment typically can be finished in seconds). Another reason for being widely used is that *BLAST* gives an expectation value for an alignment, which estimates how many times one expects to see such an alignment occur by chance. This allows a user to quantitatively assess the significance of the alignment. Although it may not be as sensitive as many other tools, *BLAST* captures most of the possible matches that have good confidence levels, and makes large-scale sequence comparisons more feasible.

2.2 Multiple Sequence Alignment

A multiple sequence alignment aligns several sequences to obtain the best commonality among them. It is the foundation for identification of functionally important regions, building sequence profile for further sequence search, protein family classification, phylogenetic reconstruction, etc. The conserved regions (motifs) in multiple sequence alignment often have biological significance in terms of structure and function. A correlated mutation between two residue positions can be used to predict a probable physical contact in structure [15] using programs such as *WHATIF* [16]. A profile derived from multiple sequence alignment is often more sensitive with less noise than the information provided by a single sequence when searching for related proteins. However, it is not realistic to use a rigorous algorithm for an alignment of more than three sequences of typical protein sizes (around 300 residues) due to its computing time. Hence, approximations have to be used in practical multiple sequence alignment tools. Active research is ongoing for this problem [17]. Like pairwise sequence alignment, multiple sequence alignment can also be categorized into global alignment and local alignment.

A widely used algorithm for global alignment is the progressive method [18]. It first aligns all possible pairs of sequences, and uses the pairwise similarity scores to construct a tree. Then it traverses the nodes of the tree, and repeatedly aligns the child nodes, i.e., sequences at the tips of the tree or clusters of aligned sequences. Once two sequences or clusters have been aligned, their relative alignment is no longer changed. Clusters of previously aligned sequences are treated as a linearly weighted profile when they are subsequently aligned with another sequence or cluster. This algorithm has been implemented in *CLUSTAL* [19], the most popular program for global multiple sequence alignment. The *GCG* program *PILEUP* [12] also uses a similar algorithm. The major difference between the two programs is in the pairwise alignment methods: *PILEUP* uses the dynamic programming algorithm [9], while *CLUSTAL* allows a user to choose between the dynamic programming algorithm and an algorithm [20] that is less sensitive but much faster. Several variants of the progressive algorithm have also been developed. *MALI* [21] is based on heuristics that search for a subset of sequence segments which are common between the sequences. *PIMA* [22] takes advantage of secondary structure prediction to weigh gap penalties while making the progressive alignment. New methods other than the progressive algorithm have been explored. For example, the *CypData* package [23] uses an iterative algorithm to generate a multiple sequence alignment by making the alignment, protein/gene tree, and pair weights mutually consistent.

Local multiple sequence alignment focuses on short similar regions across the different sequences. Most algorithms for this purpose only look for ungapped alignments, referred to as *blocks*. *MACAW* [24] is a semi-manual program, which allows a user to choose the sequences and regions in which to search for blocks during the alignment. *MEME* [25] requires a user to specify the number of blocks that are expected to occur. The occurrence of blocks defined by *MEME* is not necessarily in the same order in different sequences. Both *MACAW* and *MEME* provide statistical significance estimates for each block. The BlockMaker program [26] is fully automatic, and provides a convenient way to detect useful motifs in a family of sequences without using human inspection. It assumes all sequences contain all blocks. If a block is not found in some sequences, either the block or the sequences will automatically removed from the alignment. However, BlockMaker requires the blocks to be in the same order in all sequences.

2.3 Sequence Family and Domain Parsing

Protein sequences can be classified into families based on multiple sequence alignment. A family relationship often indicates a structural, functional, and evolutionary relationship. Different methods for multiple sequence alignment produce alternative ways to classify protein sequences into families and to align the members of a family. Depending on the need of a user, protein family classification can be based on either the alignment of long sequence

domains (typically 100 residues or more) or small conserved motifs. The former tends to be more reliable but less sensitive than the latter when using default setting of most programs.

Several methods based on sequence similarity focus more on the alignment of long sequence domains, including *Pfam* [27], *ProDom* [28], *SBASE* [29], and *COG* [30]. These methods differ in their techniques to construct families. *Pfam* builds multiple sequence alignments of many common protein domains using hidden Markov models. The *ProDom* protein domain database consists of similar domains based on recursive *PSI-BLAST* searches (*PSI-BLAST* will be discussed in the following). *SBASE* is organized through *BLAST* neighbors and grouped by standard protein names that designate various functional and structural domains of protein sequences. *COG* aims towards finding ancient conserved domains through delineating families of orthologs across a wide phylogenetic range.

Some protein sequence classifications are based on "fingerprints" of small conserved motifs in sequences, such as *PROSITE* [31], *PRINTS* [32], and *BLOCKS* [33]. In protein sequence families, some regions have been better conserved than others during evolution. These regions are generally important for protein functions or for the maintenance of 3D structures, and hence, are suitable as fingerprints. *PROSITE* and *PRINTS* derive fingerprints from gapped alignment, while *BLOCKS* contain multiply aligned ungapped segments corresponding to the most highly conserved regions of proteins. A fingerprint in *PRINTS* may contain several motifs of *PROSITE*, and thus, may be more flexible and powerful than a single *PROSITE* motif. Therefore, *PRINTS* can provide a useful adjunct to *PROSITE*.

Other protein family classifications based on sequence similarity are derived from multiple sources. The *ProClass* database [34] is a non-redundant protein database organized according to family relationship as defined collectively by *PROSITE* patterns and *PIR* superfamilies. The *MEGACLASS* server [35] provides classifications by different methods, including *Pfam, BLOCKS, PRINTS, ProDom, SBASE*, etc.

A by-product of the family classification is domain parsing, i.e., the prediction of the range of a sequence segment that forms a functional or structural domain. Such information is particularly useful in the NMR-based structure determination, which often cuts a large protein into several structurally compact domains and solves the structure of each domain separately. A family of domains from different proteins often indicates these domains have a unique function or compact structure, although the domain boundaries usually cannot be determined exactly. Among various protein family classifications, the *ProDom* and *DOMO* [36] servers are particularly effective for domain parsing.

2.4 Phylogenetic Classification

Phylogenetic relationships among proteins in different organisms may be inferred from the protein sequences. The basic idea is that the more mutations required to change one protein sequence into the other, the more unrelated the sequences and the lower the probability that they share a recent common ancestor sequence. A tree structure of proteins can be used to describe the evolutionary relationship among a family of proteins. There are different ways of measuring the "genetic distance" of proteins, and hence different types of protein trees can be constructed. Among the popular ones are *minimum distance*, *maximum parsimony*, and *maximum likelihood* trees. A minimum distance method predicts the phylogenetic relationship by constructing a protein tree to minimize the total pairwise sequence distance (i.e., the editing distance measured by the similarity between the two sequences) of adjacent tree nodes. Both maximum parsimony and maximum likelihood methods are based on multiple sequence alignments of the given protein sequences. A maximum parsimony method builds a tree to minimize the total number of evolutionary changes between proteins adjacent in the tree, while a maximum likelihood method tries to maximize the total likelihood of making such changes. A number of computer tools available for protein tree constructions. Among them are *TOPAL* [37] (minimum distance method based), *Hennig86* [38] (maximum parsimony method based), and *PAML* [39] (maximum likelihood method based). Some programs provide options to use any of the three methods, e.g., the two widely used packages *PHYLIP* [40] and *PAUP* [41].

2.5 Search Based on Multiple Sequence Alignment

One can detect remotely related proteins using the result of a known multiple sequence alignment as query. Pairwise sequence alignments require relatively high level of sequence identity (typically 25% or more) for reliable results. The characteristics in a multiple sequence alignment can

Table 2. Selected Sequence Annotation Tools

Hydrophobic Profile		
Johns Hopkins's Server	grserv.med.jhmi.edu/~raj/MISC/hphobh.html	server
Weizmann'sServer	bioinformatics.weizmann.ac.il/hydroph/	server
Transmembrane Segment Prediction		
MEMSAT	ftp.biochem.ucl.ac.uk/pub/MEMSAT/	executable
SOSUI	www.tuat.ac.jp/~mitaku/adv_sosui/	server
TMAP	www.embl-heidelberg.de/tmap/tmap_info.html	server
TMpred	ulrec3.unil.ch/software/TMPRED_form.html	server
TMHMM	130.225.67.199/services/TMHMM-1.0/	server
Motifs		
I-sites	ganesh.bchem.washington.edu/~bystroff/Isites/	server
MOTIF	www.motif.genome.ad.jp	server
Signaling Site		
DictyOGlyc	genome.cbs.dtu.dk/services/DictyOGlyc/	server
NetOGlyc	genome.cbs.dtu.dk/services/NetOGlyc/	server
NetPicoRNA	genome.cbs.dtu.dk/services/NetPicoRNA/	server
PSORT Server	psort.nibb.ac.jp:8800/	server
SignalP	www.cbs.dtu.dk/services/SignalP/	server
Secondary Structure Prediction		
PSA	bmerc-www.bu.edu/psa/	server
BTPRED	www.biochem.ucl.ac.uk/bsm/btpred/	server
Jpred	circinus.ebi.ac.uk:8081/	server
NNPRED	www.cmpharm.ucsf.edu/ nomi/nnpredict.html	server
PHD	dodo.cpmc.columbia.edu/predictprotein/	server
IBCP Server	pbil.ibcp.fr/NPSA/npsa server.html	server

significantly increase the underlying signal while reducing noise, and hence often times, a much lower level of sequence identity (as low as 15%) is needed to detect remote homologs in sequence databases.

Some search methods use sequence profiles based on a position-specific score matrix derived from a multiple sequence alignment on the similar sequences. For example, the *PSI-BLAST* program [14] searches a protein database using the profile of similar sequences found by *BLAST*. The search is carried out iteratively until a satisfactory match (e.g., a match that can derive the function or the structure of the query protein) is found or the search is converged (typically 3-4 iterations in total). At each iteration, the position-specific score matrix is updated using the new sequences in addition to the sequences found in previous iterations. Another sequence profile search engine is the *ISREC Profilescan* server [42], which aligns

a query sequence to the pre-determined profile library derived from *PROSITE* and *Pfam.*

Another type of search method based on multiple sequence alignment employs hidden Markov models (HMM) [43]. This type of method typically consists of the following three steps: (1) a standard sequence-based search to find matches for a query sequence; (2) construction of an HMM model based on the alignments between the query sequence and its matches to describe the position dependent amino acid (including deletion and insertion) probability distributions; (3) use of the result to search sequence databases to find matches to the constructed HMM model. Several computer packages based on HMM are available for sequence comparison, such as *SAM-T98* [44, 45] and *HMMER* [46].

Both *PSI-BLAST* and *SAM-T98* are widely used. *PSI-BLAST* is very fast. Typically, the results of each iteration are returned from the Web server in seconds. *PSI-BLAST* also allows users to select parameters and proteins for building sequence profiles interactively. Such a flexibility often yields more remote homologs being found. On the other hand, *SAM-T98* is slower but more sensitive. It has been shown that *SAM-T98* detects more remote homologs and generates fewer false positives at any level of true positives than *PSI-BLAST* [47]. *SAM-T98*, as an email server, does not allow interactive selection of parameters and proteins for building sequence profiles during a search process, as does *PSI-BLAST*. Users can do the search using both *PSI-BLAST* and *SAM-T98* and compare the results when any uncertainty exists.

3 SEQUENCE ANNOTATION

In this section, we will address the methods that assign and predict properties for a query sequence, including hydrophobic profile, prediction of transmembrane region, active site, and signaling sites, as well as prediction of secondary structure and solvent accessibility. These methods are based on the properties of the amino acids in a query sequence or a match between a query sequence and the characteristics obtained by sequence comparison.

3.1 Hydrophobicity Profile and Transmembrane Region Prediction

A hydrophobicity profile is derived from the hydropathy scales of the amino acids along a protein sequence. Hydropathy scale is a physichemical property that quantifies the hydrophobicity of an amino acid. Several sets of hydropathy scales are available [48, 49]. A hydrophobicity profile can be used to predict an interaction site on the surface of a globular protein, particularly for some active sites involving many charged residues [50]. For example, a highly hydrophilic region of an antigen is likely to be in an antigenic site that interacts with an antibody. It can also predict a protein's transmembrane regions, which are highly hydrophobic. The value of the hydrophobicity profile at a sequence position is obtained by averaging the hydropathy scales of several neighboring residues to reduce fluctuations. The choice of window size depends on the particular problem. A window size is suggested to be 7-9 residues for predicting surface sites, and 19 residues for predicting transmembrane regions [51]. Hydrophobicity profile plots are available in several commercial protein modeling packages, such as the *GCG* package [12] and the *Insight-II* package [52]. They can also be obtained from on-line servers, such as the *Protein Hydrophilicity/Hydrophobicity Search and Comparison Server* [53].

Several specialized tools for predicting transmembrane regions have been developed based on hydrophobicity profiles and other characteristics of transmembrane regions, e.g., aromatic residues are clustered near the interface of the transmembrane helices and proline residues are more frequent in transmembrane regions. In addition, these tools apply more sophisticated methods to enhance sensitivity. For example, *TMAP* [54] uses information derived from multiple sequence alignments and *TMHMM* [55] employs a hidden Markov model to locate transmembrane regions. Because of the strong pattern in membrane protein sequences, the predictions of transmembrane regions are generally very reliable. Since membrane protein structures are hard to obtain through experimental approaches, the prediction of transmembrane regions provides a very useful tool to study the structures of membrane proteins.

3.2 Search of Possible Active Sites

Potential active sites can be searched using the patterns extracted from motif databases such as *PROSITE* and *PRINTS* [32]. Some patterns are related to known protein functions. Hence, a match to a pattern may suggest a function of the query protein. However, since the statistical significance of a match is often low, given the few positions

involved in a pattern, a hit in databases may be a false positive. Therefore, the search results should only be used as suggestions for possible active sites. If a user knows the function of the query protein and the active site pattern involved, a search may identify the location of the active site. One can use the *MOTIF* search engine [56] for active site search, which includes *PROSITE, BLOCKS, ProDom,* and *PRINTS.*

3.3 Prediction of Signaling Sites

Signaling sites in signaling proteins often show special patterns within the sites and at the boundaries of the sites. Several Web servers employ the patterns to detect signaling sites for a query sequence. The widely used *SignalP* server [1] predicts signal peptides in secretory proteins and their cleavage sites using a neural network approach. A number of related servers have been developed by the same research group using neural networks: e.g., the *NetPicoRNA* server [57] for cleavage site analysis in picornaviral polyproteins and the *NetOGlyc* server [58] for predicting of the O-glycosylation sites of mammalian proteins. Another Web server for predicting signal peptides and domains is *SMART* [59]. *SMART* is based on the patterns derived from a collection of multiple sequence alignments, which represent more than 250 signaling and extracellular domains/sites.

3.4 Secondary Structure Prediction

Secondary structure prediction in three states (α-helix, β-sheet, and coil) from sequence has reached an averaged accuracy of more than 70% [60, 61]. Owing to this reliability, secondary structure prediction is widely used and incorporated into many other modeling tools, such as tertiary structure prediction. Early methods used simple statistical preference of each amino acid in different secondary structure types [62]. New methods, such as nearest neighbor approach [63], neural networks [64], and the utilization of multiple sequence alignments [65], have improved prediction performance significantly. The most widely used secondary structure prediction program is *PHD* [60], which uses neural networks and multiple sequence alignments. The *PSA* Server [66] provides nice graphic outputs for the probability of each secondary structure type along the sequence. *I-sites* [67] predicts local structures, which may include several contiguous secondary structures, using a set of sequence patterns that strongly correlate with protein structure on the local level. The *SOSUI* server [68] specializes the secondary structure prediction of membrane proteins with high accuracy. The *Consensus Secondary Structure Prediction Server* [69] gives predictions using different methods, such as *SOPM* [70], *DSC* [71], *PHD*, and *PREDATOR* [72], and builds a consensus from them. Some secondary structure prediction programs, such as *PHD* and the *PSA* Server, also predict solvent accessibility of each residue on a sequence, i.e., whether it is buried in the interior of the structure or on the surface.

Figure 1 describes a partial output from the consensus server for the secondary structure prediction of the protein cyanase. As an example, it does not represent the general performance of different programs, but it shows typically what can be expected from secondary structure prediction. One can see that the secondary structure locations are basically predicted correctly by all the programs. However, none of the programs predicts the boundaries of the secondary structures accurately. The prediction performance varies from protein to protein. In some cases, the secondary structure type or the location of a secondary

Fig. (1). Secondary structure predictions for the first 80 residues of cyanase (156 residues in total) using the *Consensus Secondary Structure Prediction Server* [69]. The protein sequence, prediction results from nine methods, and the secondary structure assignment using *DSSP* [83] based on the experimental structure (labeled by "ACTUAL" and shaded) are shown. The "h", "e", and the blank space are the predictions of α-helix, β-sheet, and loop conformation, respectively.

structure can be predicted incorrectly. Secondary structure predictions for small proteins (with less than 100 residues), especially those having several disulfide bonds, are usually poor.

Some programs focus on the content of secondary structures (the percentage of helix, strand, and coil in a protein). They generally have higher accuracies for the content of secondary structures than secondary structure prediction programs. The *SSCP* server [73, 74] uses neural networks to predict the content of secondary structures based on the amino acid composition as the only input information.

4 STRUCTURE ANALYSIS

In this section, we will discuss the modeling tools for analysis based on protein structures obtained through experimental approaches or structure predictions. These tools cover a wide range of methods, including structure visualization, geometry analysis, structure comparison, structure-based family, molecular dynamics, quantum mechanics, and electrostatics.

4.1 Visualization

Visualization is often the first step to inspect a structure. Through different display methods, e.g., ribbons, molecular surface, cartoon, and lines, structure visualization provides a convenient way to study spatial relationships of atoms, residues, secondary structures, domains, and subunits. Commercial packages for protein modeling, such as *Insight-II* [52], *SYBYL* [75], and *LOOK* [76], typically include visualization tools with extensive features. Users can also find popular public domain visualization tools, such as *VMD* [77] and *RasMol* [78]. Several tools are best known for their unique strengths for particular visualization aspects. *Molscript* [79], which produces illustrative graphs in postcript format with high quality, is widely used by researchers in their publications. *CHIME* [80] shows protein graphics inside Web browsers. *TOPS* [81] can automatically generate protein topology cartoons, using circles and triangles to depict the arrangement of α-helices and β-strands. *GRASP* [82] can show protein surface color-coded with electrostatic potential or geometry properties.

4.2 Geometry Analysis

Geometry analysis of a given protein structure provides further information related to the conformation and energetics, as well as the quality of a structure model. There are two types of geometry analysis. One is based on the geometrical relationship between atoms. For example, *DSSP* [83] is a program that assigns protein secondary structure based on the geometrical features of the hydrogen bonds on protein backbones; *HBPLUS* [84] determines a hydrogen bond according to the atomic distances and angles. Another type of geometry analysis is based on solvent-accessible surface [85] and molecular surface [86]. The two types of surfaces are defined through an imaginary spherical probe (as a model for a water molecule) with a typical radius of 1.4 Å rolling on the protein structure while maintaining contact with the van der Waals surface of the protein. The trace of the probe center is the solvent-accessible surface, while the inward-facing surface of the probe sphere as it rolls over the protein is the molecular surface. Solvent-accessible surface area can be calculated using *NACCESS* [87] or ASC [88]. The *Molecular Surface Package* [89] can compute the molecular surface area and volume. One can use hydrophobic and hydrophilic surface areas to derive semi-empirical energetics [90, 50, 91], such as solvation energy, entropy, and free energy in protein folding or binding. Another application of protein surface is domain partitioning, which cuts a protein structure into several compact domains measured by their surface area and volume. The *Protein Domain Server* [92] can be used for domain partitioning. In addition, the *DALI* domain library [93]) and the *3Dee* database [94] provide the domain definitions for the structures in the PDB [95].

Geometry analysis can also be employed to check the quality of a protein structure model. Various errors can be generated when building a structure model, including (a) bad backbone conformations, e.g., artificial *cis* peptide bonds; (b) poor stereochemistry, e.g., unwanted D-amino residues; and (c) unfavorable inter-residue packing. These errors can be detected using programs such as *WHATIF* [16] and *PROCHECK* [96]. The overall quality of a model can be further assessed by *PROVE* [97], which checks the departures of the assessed structure from the standard atomic volumes in high quality experimental structures.

4.3 Structure Comparison and Structure Family

The 3D structures of proteins are better conserved during evolution than their sequences. Two proteins can share a similar structural fold even when their sequences are not similar, and in some cases not homologous. The relationship

Table 3. Selected Structure Annotation Tools

Visualization		
CHIME	www.mdli.com	Web
gOpenMol	laaksonen.csc.fi/gopenmol/gopenmol.html	executable
GRASP	trantor.bioc.columbia.edu/grasp/	executable
LOOK	www.mag.com	executable
RasMol	klaatu.oit.umass.edu/microbio/rasmol/	executable
VMD	www.ks.uiuc.edu/Research/vmd/	executable
Geometry Analysis		
HBPLUS	www.biochem.ucl.ac.uk/mcdonald/hbplus/	executable
NACCESS	sjh.bi.umist.ac.uk/naccess.html	executable
WAHTIF	www.sander.embl-heidelberg.de/whatif/	executable
Domain Partition		
3Dee	circinus.ebi.ac.uk:8080/3Dee/help/help_intro.html	server
Domain Server	bonsai.lif.icnet.uk/domains/assign.html	executable
Alignment / Family		
SCOP	scop.mrc-lmb.cam.ac.uk/scop/	server
CATH	www.biochem.ucl.ac.uk/bsm/cath/	server
CE	cl.sdsc.edu/ce.html	server
Dali Domain Dictionary	columba.ebi.ac.uk:8765/holm/ddd2.cgi	server
FSSP	www2.ebi.ac.uk/dali/fssp/	server
HOMSTRAD	www-cryst.bioc.cam.ac.uk/~homstrad/	server
HSSP	swift.embl-heidelberg.de/hssp/	server
LPFC	bioinfo.mbb.yale.edu/align/	server
VAST	www.ncbi.nlm.nih.gov/Structure/VAST/	server
Molecular Dynamics		
AMBER	www.amber.ucsf.edu:80/amber/	executable
CHARMM	yuri.harvard.edu/charmm/charmm.html	executable
GROMOS	igc.ethz.ch/gromos/	executable
NAMD	www.ks.uiuc.edu/Research/namd/namd.html	executable
TINKER	dasher.wustl.edu/tinker/	executable
X-PLOR	xplor.csb.yale.edu/xplor-info/xplor-info.html	executable

between proteins having similar folds is clearly revealed through structure-structure comparison, which often provides more reliable information about the relationship between proteins than sequence-sequence comparison alone. Several structure comparison tools are available, e.g., *VAST* [98], *SARF* [99], and *ProSup* [100]. A popular tool for comparing a query protein structure against all the structures in the PDB is the *DALI server* [101]. When new structures are solved, researchers often submit them to the *DALI* server to find structural neighbors and their alignments. The results may reveal biologically interesting

similarities that are not detectable by sequence comparison.

The relationship between the proteins in a structure database can be classified at different hierarchical levels according to structural and evolutionary relationships. A widely used classification includes family, superfamily, and fold [102]. Proteins clustered into a family are clearly evolutionarily related with a significant sequence identity between the members. Different families whose structural and functional features suggest a common evolutionary origin are placed together in a superfamily. Different superfamilies are categorized into a fold if they have the same major secondary structures in the same arrangement and with the same topological connections. The structural similarities between different superfamilies in the same fold may arise just from the protein energetics favoring certain packing arrangements instead of a common evolutionary origin. Most protein structure classification tools follow the concepts similar to family, superfamily, and fold, but differ due to detailed classification criteria and different structure-structure comparison methods. *CATH* [103] is a hierarchical classification of protein domain structures. *CE* [104] provides structural neighbors of the *PDB* entries with structure-structure alignments and 3D superpositions. *FSSP* [105] features fold tree, sequence neighbors, and multiple structure alignments. *SCOP* uses augmented manual classification with the hierarchical levels of class, fold, superfamily, and family of close homologs [102]. Among them, *SCOP* provides more function related information. However, *SCOP* is not updated as frequently as others due to the manual work involved, while *FSSP* and *CATH* follow the *PDB* updates closely.

4.4 Molecular Dynamics, Quantum Mechanics, and Electrostatics

Most protein functions are achieved through a dynamic process. A well established method to study a dynamic process of protein is molecular dynamics simulation [106, 107], which has been applied to proteins for more than two decades [108, 109, 110]. A molecular dynamics simulation uses a given structure for the initial coordinates. Each atom is modeled as a particle with a certain mass and a partial charge. The force fields, which describe atomic interactions such as bond energy, van der Waals energy, and Coulomb energy, are based on empirical functions with analytical forms. Several sets of energy function parameters have been developed, including *CHARMM* [111], *GROMOS* [112], and *AMBER* [113]. After assigning random initial velocities to the atoms of the protein according to the Boltzman distribution for a given temperature, the dynamics governed by the Newton's Law are carried out using numerical integrations with a time step of about one femtosecond ($1X10^{-15}$ second). Many molecular dynamics simulation programs are available, such as *CHARMM*, *GROMOS*, and *AMBER*, *TINKER* [114], *XPLOR* [115] and *NAMD* [116]. A molecular dynamics simulation can be used to study small conformational change and energetics such as free energy differences between two protein states. A limitation of molecular dynamics simulation is that the time scale it can model (up to several nanoseconds for a sizable protein) is shorter than many interesting dynamic processes in protein (at a time scale of several seconds or longer). Active research is going to reach longer time scales through algorithm developments [117, 118], parallel implementations [119, 120], and special protocols to artificially accelerate a dynamic process [121, 122].

Classical molecular dynamics simulation alone cannot describe the quantum mechanical processes, such as electronically excited states, spectroscopic transitions, and chemical reactions in which bonds are altered. The modeling tools for quantum mechanical calculations, such as *GAUSSIAN* [123], *GAMESS* [124], and *Q-Chem* [125], are designed to tackle these problems. They can also be used to obtain atomic partial charges and parameters of energy functions for molecular dynamics simulation. However, the quantum mechanics calculation is very time consuming to simulate a whole protein. A good approach is to combine a quantum mechanical treatment for a small part of a system with a molecular dynamics simulation procedure to the rest [126, 127, 128]. This allows the description of processes which cannot be represented by a molecular dynamics potential.

Another weakness of classical molecular dynamics simulation is the description of solvation effects, such as solvation energy and electrostatics. Although molecular dynamics simulation can add explicit water molecules around a protein, it is often insufficient to describe solvation effects due to the lack of description for electronic polarization and the limited time scale it can simulate. A better way to calculate solvation effects is to use continuum electrostatics [129, 130] governed by the Poisson-Boltzman equation, where the water is modeled by continuum media with a dielectric constant of about 80. A widely used program is DelPhi [131], which uses finite difference method to solve the Poisson-Boltzmann equation.

5 PREDICTION OF 3D STRUCTURE

Predicting the 3D structure of a protein from its amino acid sequence using computational methods becomes more and more practical due to the development of new methods. Many non-trivial structure predictions [132, 133, 134] produced prior to the experimental structure determinations turned out to be fairly accurate. Most notably, the success of protein structure prediction has been demonstrated in the community-wide experiments in the Critical Assessment of Techniques for Protein Structure Prediction (CASP) [135, 136, 137]. In this contest, there are two types of tertiary structure predictions, i.e., *ab initio* methods which predicts a protein structure based on physi-chemical principles directly, and template-based methods, which use known protein structures as templates. Template-based methods include homology or comparative modeling, and fold recognition via threading. The coverage of protein sequences by template-based methods (about 50-70% now) is expanding as more and more structures are solved.

5.1 Homology Modeling

Homology modeling constructs the coordinates of all the atoms in a query protein based on sequence alignment between the query protein and another protein of known 3D structure. It typically consists of three steps: (1) identify the protein templates with known 3D structures and produce an alignment between the query sequence and its templates; (2) build the model for the query protein given its alignment with the template structures; (3) evaluate the quality of the model.

A conventional homology modeling requires a high sequence identity between a query protein and its template in the protein structure database PDB [95] for a reliable template recognition and the sequence alignment. However, a sequence search based on multiple sequence alignment can also be used to find a suitable template, and often produces better results than pairwise sequence alignment, as shown in CASP-3 [137]. Template search and alignment are essential for the correctness and the quality of a homology model. Homology modeling programs always generate a structure for any query sequence using the conformation of the template structures and the alignments between the query protein sequence and its templates. If the templates or the alignments are incorrect, the output model will certainly be wrong as well.

Different homology modeling methods use different approaches to construct a 3D model from given templates and alignments. One way to construct an atomic model is to use only the backbone coordinates from the template, and to build sidechain independently with tools such as *SCWRL* [138] and *MaxSprout* [139]. Alternative methods for constructing atomic models employ sidechain conformations of templates as well. Automated servers (e.g., *SWISSMODEL* [140] and *CPHmodels* [141]) provide an interface to submit a sequence and get the model either interactively or through email. These servers are fast and easy to use. The *WHATIF* program [16] provides the option to construct a crude model quickly or to build a structure using a better, but much slower method (several hours for a large protein). *COMPOSER* [142] has a specific tool to deal with the loop regions which contain gaps in the alignment. *COMPOSER* under *SYBYL* [75] also provides an interactive Graphic User Interface (GUI) for model building, which allows a user to edit at each step. The most widely used homology modeling program is *MODELLER* [143]. It starts with an extended strand for the query protein, and then folds it to satisfy spatial restraints derived from the alignment between the query sequence and its templates. In particular, it tries to preserve main chain dihedral angles or hydrogen bonding features from the template structures. *MODELLER* also uses physical force fields to prevent atoms from clashing with each other. In the loop regions, with gaps in the alignment, *MODELLER* uses statistical information derived from the alignment of many proteins of known 3D structure. The final 3D model is obtained by optimization through conjugate gradients and molecular dynamics with simulated annealing.

The quality of a model depends primarily on the sequence identity between the query protein and the template. The higher the sequence identity, the more accurate the structure derived from homology modeling. For high sequence identity (typically 40% or more), it is not rare that homology modeling produces models with an all-atom RMSD lower than 2 Å between the model and the experimental structure. Fig. 2(a) shows an example for the typical quality of a constructed model. A challenge in homology modeling is the construction of regions with large alignment gaps. Although loops with short alignment gaps can often be modeled successfully, insertions of about 8 residues or more in the query sequence usually cannot be modeled reliably. It is important to use the quality assessment tools to check the structure model. If errors are found, one can adjust the alignment and rebuild the model. Another method to use is to generate multiple models and find the model with the least errors. It may be necessary to repeat the process of alignment, model construction,

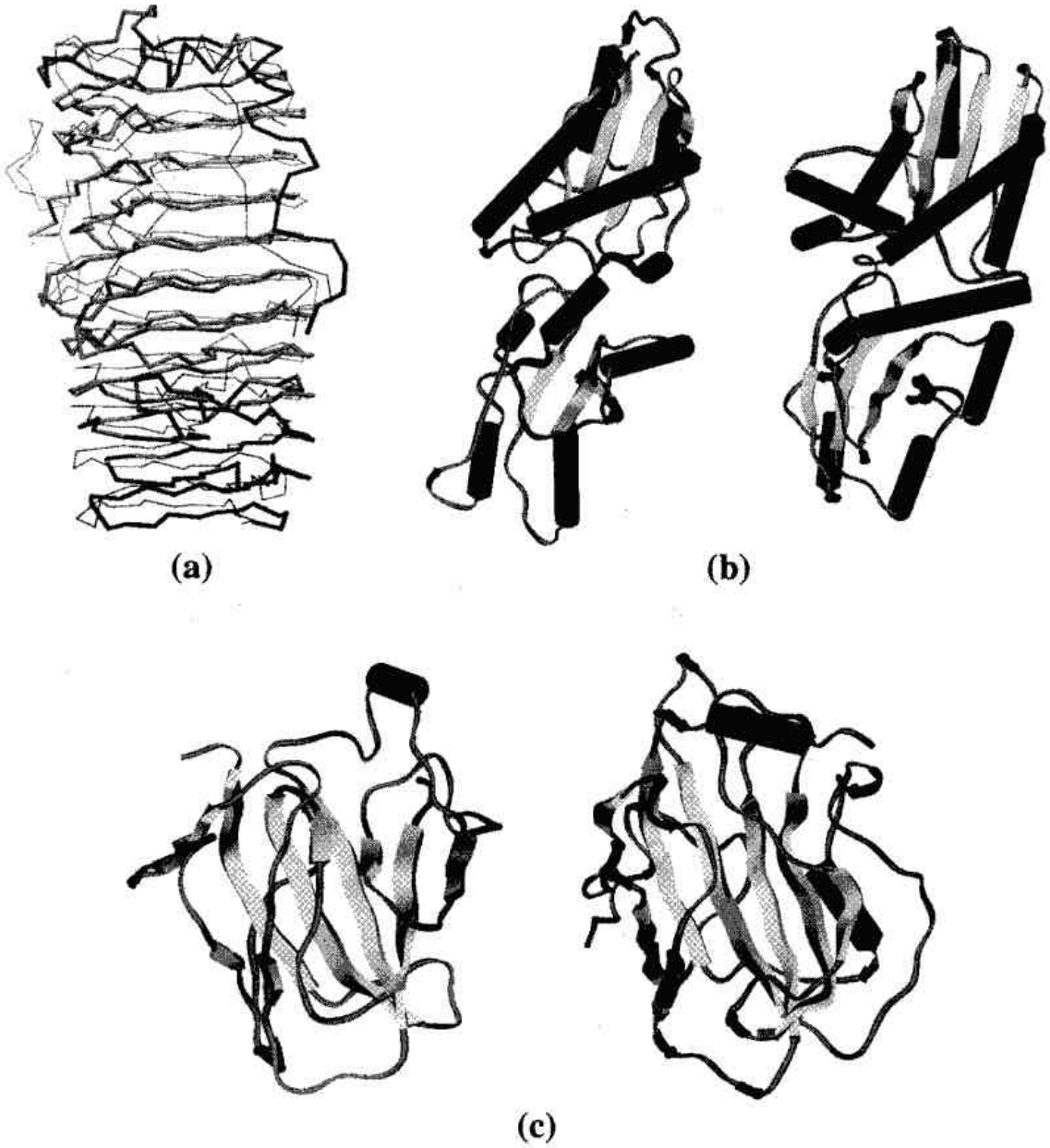

Fig. (2). Comparisons between the predicted models using *PROSPECT* [147] and the experimental structures in the CASP-3 [180]. (a) Target t0068 drawn by VMD [77]. The predicted model is in thick lines and the experimental structure is in thin lines. The template used and the target protein belong to the same family with the sequence identity of 25%. One can see that almost all the backbone structures superimpose well between the model and the experimental structure. (b,c) Targets t0053 and t0067, respectively, drawn by Insight-II [52]. The predicted models are at the left and the experimental structures are at the right. The cylinders indicate alpha-helices, the strands indicate beta-sheets, the dark lines indicate turns, and the thin lines indicate loops. The templates used and the target proteins belong to the same superfamily for t0053, and the same fold for t0067, neither with significant sequence identity. The predicted models for t0053 and t0067 provide good folds but some portions of the backbones in the models have wrong conformations.

and assessment until a satisfactory model is obtained.

5.2 Threading

Protein threading (sequence-structure alignment) [144, 145, 146, 147, 148] is a promising template-based method for fold recognition, which identifies a suitable fold from a structure library for the query sequence and provides an alignment between the query protein and the fold. The basic idea of threading can be summarized as follows. Given a query protein sequence s of unknown structure, threading searches the structure templates **T** to find

Table 4. Selected Protein Structure Prediction Tools

Homolgy Modeling		
COMPOSER	www-cryst.bioc.cam.ac.uk/ www.tripos.com/software/composer.html	executable module (GUI)
CONGEN	www.congenomics.com/congen/congen_toc.html	executable
CPHmodels	www.cbs.dtu.dk/services/CPHmodels/	server
DRAGON	www.nimr.mrc.ac.uk/~mathbio/a-aszodi/dragon.html	executable
LOOK	www.mag.com/products/look.html	module (GUI)
MODELLER	guitar.rockefeller.edu/modeller/ www.msi.com/solutions/products/insight/modules/Modeler.html	executable module (GUI)
SWISS-MODEL	www.expasy.ch/swissmod/SWISS-MODEL.html	server
WHAT IF	www.sander.embl-heidelberg.de/whatif/	executable
Singleton Threading		
123D	www-lmmb.ncifcrf.gov/~nicka/123D.html	server
TOPITS	dodo.cpmc.columbia.edu/predictprotein/	server
SAS	www.biochem.ucl.ac.uk/bsm/sas/	server
UCLA-DOE	www.doe-mbi.ucla.edu/people/frsvr/frsvr.html	server
Threading Using Pairwise Interactions		
NCBI Package	www.ncbi.nlm.nih.gov/Structure/	executable
PROFIT	lore.came.sbg.ac.at/	executable
PROSPECT	compbio.ornl.gov/structure/prospect/	executable
THREADER	globin.bio.warwick.ac.uk/~jones/threader.html	executable
ToPLign	cartan.gmd.de/ToPLign.html	server
Docking		
AutoDock	www.scripps.edu/pub/olson-web/doc/autodock/	executable
DOCK	www.cmpharm.ucsf.edu/kuntz/dock.html	executable

the best fit for **s**. A threading requires four components [149]: (1) a library **T** of representative 3D protein structures for use as templates; (2) an energy function to describe the fitness of any alignment between **s** and **t**, where t is a template in **T**; (3) a threading algorithm to search for the lowest energy among the possible alignments for a given **s-t** pair; (4) a criterion to estimate the confidence level of the predicted structure. The threading approach can be further subdivided into two categories: (1) threading that considers only the preference of amino acids in the query sequence at single sites of the templates (singleton threading); (2) threading that uses the preference on pairs of amino acids in the query sequence within a contact distance when they are aligned to a given structure. In general, singleton threading is faster, while threading using pairwise interactions is more sensitive to detect the correct templates.

Singleton threading constructs a one-dimensional (1D) structure profile for each residue position in a template structure using local 3D environmental information such as secondary structure type, degree of environmental polarity, and the fraction of the residue surface accessible to

solvent. The energy function is based on the compatibility of the 20 amino acids for each position in the 1D structure profile. The compatibility is derived from the statistics of the whole template database. Optimal 1D alignments between a query sequence and a template can be determined by dynamic programming. The final template is selected according to the optimal score or its statistical significance. The singleton threading can incorporate secondary structure predictions and position-dependent profiles based on multiple sequence alignments into the energy function. Several servers are available for singleton threading, e.g., *123D* [150], *TOPITS* [151], *SAS* [152], and the *UCLA-DOE Structure Prediction Server* [153].

Threading using pairwise interactions considers the propensity of two amino acids in the target sequence to be aligned within a specified distance using a score function compiled from a database of structures. In the recent CASP-3, top performers were most often among the groups using threading with pairwise interactions [137]. Several threading programs using pairwise interactions are available, including the *NCBI Threading Package* [154], *PROFIT* [145], *PROSPECT* [147], and *THREADER* [146]. The *NCBI Threading Package* provides a good statistical assessment for threading result. *PROSPECT* guarantees to find the globally-optimal alignments for a given energy function with pairwise interactions. Figure 2 (b,c) shows the prediction results for two CASP-3 targets using *PROSPECT*. It provides a typical example of structure information that can be expected from successful threading.

The threading approach is more sensitive than the sequence-based search methods like *PSI-BLAST* and *SAM-T98*. However, a key difficulty for threading is that the structure profile and the residue pairs derived from the template may not adequately describe the corresponding information in the query protein due to the structure difference between the two proteins, even when they share the same fold. This is a more significant problem in the fold category than in the superfamily category. In the CASP-3, almost every protein in the superfamily category was predicted correctly by at least one threading program. However, few proteins in the fold category were predicted correctly by any method.

5.3 *Ab Initio* Prediction

An *ab initio* protein structure prediction derives a structure model through the optimization of an energy function which describes the physical properties or statistical preferences of amino acids. *Ab initio* tertiary structure prediction from sequence has proven to be extremely difficult even after tremendous effort for decades [155, 156, 157, 158, 159]. *Ab initio* prediction programs require long computing time, and the prediction results are generally unreliable. However, some recent developments using hierarchic approaches, which first build local structures and then assemble them into a global structure, seem to provide new hope for generating low resolution structures. Once local structures are more or less defined, assembling them requires a significantly smaller computational search space. The optimization process is typically carried out using genetic algorithms [160] or Monte Carlo simulations [157]. Local structures can be built through a search based on empirically derived data about preferred torsion angles in secondary structure elements as done by the program *LINUS* [161]. The "mini-threading" method [162] may be a more efficient way to build local structures. Mini-threading methods obtain the matches between short structure segments of template and the query sequence for building local structures. Some success of mini-threading has been demonstrated in CASP-3 [137]. However, *ab initio* prediction programs are typically unavailable to the general research community.

5.4 Protein Docking

Protein docking determines a bound structure complex formed from two proteins or a protein and a substrate, starting with two separate unbound structures. When the conformational changes of each structure upon binding are assumed to be insignificant (so called "rigid binding"), one can often use shape complementarity to find tight match between the surfaces of the two structures [163, 164]. In addition to the geometric fitness, the energetics across the binding interface can be also considered [165, 166]. A widely used docking program is *DOCK* [163]. Prediction of rigid binding often finds the experimental binding conformation ranked among the top of the candidate list. When a small ligand is flexible and the binding protein is rigid, the search problem to find an optimal solution can still be manageable, although the results tend to be less reliable than the rigid docking. *AutoDock* [167] is a program to predict the bound conformations between flexible ligands and rigid proteins. When the larger structure in the binding complex undergoes a significant conformational change upon binding, e.g., in some protein-protein interactions, the structure flexibility makes the induced docking problem as difficult as the *ab initio* structure prediction. Current docking

techniques are typically unable to identify the bound structure in this case.

6 DISCUSSIONS

In this Section, we discuss some general issues in protein modeling, including the availability of tools and the relationship between experimental approaches and computational methods, as well as current trends and future outlook.

6.1 Availability of Tools

Most protein tools can be used through Web servers or downloaded from the Internet. A reader can get more information about these tools through their Web pages (see Table 1-4). One can also find more tools through links at our Web page http://compbio.ornl.gov /structure/resource/. Most of the tools are free of charge or with a minimum cost to the academic users, while commercial users sometimes have to pay a fee for license. Several commercial packages for protein modeling, such as *Insight-II* [52] by the Molecular Simulations Inc., *GCG* [12] by the Genetics Computer Group, *SYBYL* [75] by the Tripos Associates, Inc., and *LOOK* [76] by the Molecular Applications Group, provide various modules for different types of protein modeling. While these packages may be expensive, they typically have friendly graphic user interfaces with few computer bugs. In addition, technical supports can be provided from the commercial vendors.

6.2 Experimental *vs*. Computational Approaches

Experimental approaches and computational methods complement each other in protein science. Modern experimental techniques rely more and more on computing. There are many computer tools that assist experimental measurement or interpret experimental data, for example, tools to help determine X-ray crystallographic structures. Many experimentalists use computational tools routinely to study proteins. On the other hand, most results from computational tools are predictions and subject to further experimental verification. A user should always keep in mind the general quality and the confidence level of the predictions when using them to draw any conclusion. Usually, it is rewarding to try different tools available. The consensus and variations among different predictions may provide a clue about whether the predictions are reliable or not. Whenever any experimental information is available, a user should incorporate the information in the tools or at least use the information to verify the output results.

6.3 Trends and Outlook

Protein modeling is a rapidly developing field, where new methods and tools are produced frequently. Several current trends, as listed below, probably indicate the future directions of this field for the next decade.

- **Web interfaces**. As shown above, a large number of tools, particularly sequence analysis tools, are implemented in the Web servers. Some servers, e.g., the Biology WorkBench [168], provide a Web-based computing environment that integrates a wide variety of analysis programs into a single interface.

- **Genome-wide analysis.** Several groups employed computational tools to study all the coding sequences in a whole genome [169, 170, 171]. These studies provide timely analyses for the current genome sequencing efforts, and allow gene-hunting researchers to find valuable information quickly. They may also help the understanding of a genome as a whole and the comparison between different species.

- **Large-scale modeling**. Using parallel/network computers and better algorithms, researchers are reaching larger and larger scales in modeling, e.g., (1) combinatorial search to find the optimal solutions for complex computing problems [147]; (2) large systems [172], particularly complex system of proteins with their environments (solvent, lipid, etc.) [173]; (3) longer time scales for molecular dynamics simulations [174].

- **Interactive modeling**. Several modeling tools allow users to provide input interactively during a modeling process [175]. For example, one can carry out a molecular dynamics simulation in an interactive computer graphics system that keeps track of user control (e.g., manually moving a water atom away from a protein) while maintaining a physically valid representation [176, 177]. Virtual reality and speech recognition as possible input methods for interactive modeling have also been explored.

- **Using a combination of tools together**. It is often more fruitful combining different modeling tools to study a particular protein problem. For example, sequence alignment, transmembrane segment prediction, secondary structure prediction, homology modeling, and molecular dynamics simulation were applied in predicting the structure of a membrane protein [133]. In another example, docking, molecular dynamics simulation, electrostatics, and quantum mechanics were used when studying the binding between a ligand and a receptor [178, 179].

6.4 Summary

In summary, significant advances during the past two decades have made protein modeling tools more reliable and easy to use. Not only computational biologists but also experimentalists benefit tremendously from these tools, which often provide useful information about the structure and function of a protein. However, one cannot use modeling tools blindly. Further experimental evidence may be needed for some predictions, which could be inaccurate or even wrong. There are still many challenging problems in protein modeling and the related research is very active. We believe, with the technical improvement in modeling methods and so many genes (protein sequences) discovered, protein modeling tools will play an even more important role in the post-genome era.

ACKNOWLEDGMENTS

We thank Dr. Michael A. Unseren for a critical reading of this manuscript. We also thank Drs. Oakley H. Crawford and J. Ralph Einstein for helpful discussions. This research was sponsored by the Office of Health and Environmental Research, U.S. Department of Energy, under Contract No. DE-AC05- 96OR22464 with Lockheed Martin Energy Research Corporation.

REFERENCES

[1] Nielsen, H., Engelbrecht, J., Brunak, S., and von Heijne, G. (**1997**). *Protein Eng., 10*, 1-6.

[2] Russo, E. (**1999**). *The Scientist, 21*, 8-8.

[3] Brutlag, D. L. (**1998**). *Curr. Opinion Microbiol., 1*, 340-345.

[4] Bairoch, A. and Apweiler, R. (**1999**). *Nucleic Acids Research, 27*, 49-54.

[5] Barker, W. C., Garavelli, J. S., McGarvey, P. B., Marzec, C. R., Orcutt, B. C., Srinivasarao, G. Y., Yeh, L. L., Ledley, R. S., Mewes, H., Pfeiffer, F., Tsugita, A., and Wu, C. (**1999**). *Nucleic Acids Research, 27*, 39-42.

[6] Dayho,, M. O. (**1978**). *Atlas of Protein Sequences and Structure, 5*(Supplement 3), 345-352.

[7] Heniko,, S. and Heniko,, J. G. (**1992**). *Proc. Natl. Acad. Sci. USA, 89*, 10915-10919.

[8] Gonnet, G. H., Cohen, M. A., and Benner, S. A. (**1992**). *Science, 256*, 1443-1445.

[9] Needleman, S. B. and Wunsch, C. D. (**1970**). *J. Mol. Biol., 48*, 443-453.

[10] Smith, T. F. and Waterman, M. S. (**1981**). *Adv. Appl. Math., 2*, 482-489.

[11] Hughey, R. (**1996**). *CABIOS, 12*, 473-479.

[12] Genetics Computer Group (**1994**). *GCG Program Manual for the Wisconsin Package*, Version 8. Genetics Computer Group, Inc., Madison, Wisconsin.

[13] Pearson, W. R. and Lipman, D. J. (**1988**). *Proc. Natl. Acad. Sci. USA, 85*, 2444-2448.

[14] Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (**1997**). *Nucleic Acids Research, 25*, 3389-3402.

[15] Gobel, U., Sander, C., Schneider, R., , and Valencia, A. (**1994**). *Proteins, Struct. Funct. Genet., 18*, 309-317.

[16] Vriend, G. (**1990**). *J. Mol. Graphics, 8*, 52-56.

[17] Gotoh, O. (**1999**). *Adv. Biophys., 36*, 159-206.

[18] Feng, D. F. and Doolittle, R. F. (**1987**). *J. Mol. Evol., 25*, 351-360.

[19] Thompson, J. D., Higgins, D. G., and Gibson, T. J. (**1994**). *Nucleic Acids Research, 22*, 4673- 4680.

[20] Wilbur, W. J. and Lipman, D. J. (**1983**). *Proc. Natl. Acad. Sci. USA, 80*, 726-730.

[21] Vingron, M. and Argos, P. (**1989**). *Comput. Appl. Biosci., 5*, 115-121.

[22] Smith, R. F. and Smith, T. S. (**1990**). *Proc. Natl. Acad. Sci. USA*, **87**, 118-122.

[23] Gotoh, O. (**1996**). *J. Mol. Biol., 13*, 823-838.

[24] Lawrence, C. E., Altschul, S. F., Boguski, M. S., Liu, J. S., Neuwald, A. F., and Wootton, J. C. (**1993**). *Science, 262*, 208-214.

[25] Bailey, T. L. and Gribskov, M. (**1998**). *J. Comp Biol., 5*, 211-221.

[26] Heniko,, S., Heniko,, J. G., Alford, W. J., and Pietrokovski, S. (**1995**). *Gene, 163*, GC17-26.

[27] Bateman, A., Birney, E., Durbin, R., Eddy, S. R., Finn, F. D., and Sonnhammer, E. L. L. (**1999**). *Nucleic Acids Research, 27*, 260-262.

[28] Corpet, F., Gouzy, J., , and Kahn, D. (**1999**). *Nucleic Acids Research, 27*, 263-267.

[29] Murvai, J., Vlahovicek, K., Barta, E., Szepesvari, C., Acatrinei, C., and Pongor, S. (**1999**). *Nucleic Acids Research, 27*, 257-259.

[30] Tatusov, R. L., Koonin, E. V., and Lipman, D. J. (**1997**). *Science, 278*, 631-637.

[31] Hofmann, K., Bucher, P., Falquet, L., and Bairoch, A. (**1999**). *Nucleic Acids Research, 27*, 215-219.

[32] Attwood, T. K., Flower, D. R., Lewis, A. P., Mabey, J. E., Morgan, S. R., Scordis, P., Selley, J., and Wright, W. (**1999**). *Nucleic Acids Research, 27*, 220-225.

[33] Heniko,, J. G., Heniko,, S., and Pietrokovski, S. (**1999**). *Nucleic Acids Research, 27*, 226-228.

[34] Wu, C., Shivakumar, S., and Huang, H. (**1999**). *Nucleic Acids Research, 27*, 272-274.

[35] States, D. J., Harris, N. L., and Hunter, L. (**1993**). *Proc. Intel. Syst. for Mol. Biol., 1*, 387-394.

[36] Gracy, J. and Argos, P. (**1998**). *Bioinformatics, 14*, 174-187.

[37] McGuire, G. and Wright, F. (**1997**). *Bioinformatics, 14*, 219-220.

[38] Farris, J. S. (**1989**). *Cladistics, 5*, 163.

[39] Yang, Z. (**1999**). *Phylogenetic Analysis by Maximum Likelihood (PAML)*. University College London, London, UK.

[40] Felsenstein, J. (**1989**). *Cladistics, 5*, 164-166.

[41] Swofford, D. L. (**1999**). *PAUP*, Phylogenetic Analysis Using Parsimony and Other Methods, Version 4*. Sinauer Associates, Sunderland, Massachusetts.

[42] Bucher, P. (**1999**). *The ISREC Profilescan Server*. The Swiss Institute for Experimental Cancer Research, Epalinges, Switzerland.

[43] Eddy, S. R. (**1996**). *Curr. Opinion Struct. Biol., 6*, 361-365.

[44] Krogh, A., Brown, M., Mian, I. S., Sjolander, K., and Haussler, D. (**1994**). *J. Mol. Biol., 235*, 1501-1531.

[45] Karplus, K., Barrett, C., and Hughey, R. (**1998**). *Bioinformatics, 14*, 846-856.

[46] Eddy, S. R., Mitchison, G., and Durbin, R. (**1995**). *J. Comp Biol., 2*, 9-23.

[47] Park, J., Karplus, K., Barrett, C., Hughey, R., Haussler, D., Hubbard, T., and Chothia, C. (**1998**). *J. Mol. Biol., 284*, 1201-1210.

[48] Kyte, J. and Doolittle, R. F. (**1982**). *J. Mol. Biol., 157*, 105-132.

[49] Engleman, D. M., Steitz, T. A., and Goldman, A. (**1986**). *Ann. Rev. Biophys. Chem., 15*, 321-353.

[50] Xu, D., Lin, S. L., and Nussinov, R. (**1997**). *J. Mol. Biol., 265*, 68-84.

[51] S. R. Krystek, J., Metzler, W. J., and Novotny, J. (**1997**). In Coligan, J. E., Dunn, B. M., Ploegh, H. L., Speicher, D. W., and Wingfield, P. T., Eds., *Current Protocols in Protein Science*, pages 2.2.1-2.2.13. John Wiley & Sons, New York.

[52] Molecular Simulations Inc. (**1998**). *Insight II* (Release 98.0). San Diego, California.

[53] Prilusky, J., Hansen, D., Pilpel, T., and Safran, M. (**1999**). *The Protein Hydrophilicity/Hydrophobicity Search and Comparison Server*. Weizmann Institute of Science, Rehovot, Israel.

[54] Persson, B. and Argos, P. (**1994**). *Journal of Molecular Biology, 237*, 182.

[55] Sonnhammer, E. L. L., von Heijne, G., and Krogh, A. (**1998**). *ISMB, 6*, 175-182.

[56] Institute for Chemical Research (**1999**). *MOTIF*. Kyoto University, Kyoto, Japan.

[57] Blom, N., Hansen, J., Blaas, D., and Brunak, S. (**1996**). *Protein Science, 5*, 2203-2216.

[58] Hansen, J. E., Lund, O., Rapacki, K., and Brunak., S. (**1997**). *Nucleic Acids Research, 25*, 278-282.

[59] Schultz, J., Milpetz, F., Bork, P., and Ponting, C. (**1998**). *Proc. Natl. Acad. Sci. USA, 95*, 5857-5864.

[60] Rost, B. and Sander, C. (**1993**). *J. Mol. Biol., 232*, 584-599.

[61] Frishman, D. and Argos, P. (**1997**). *Proteins, Struct. Funct. Genet., 27*, 329-335.

[62] Chou, P. Y. and Fasman, G. D. (**1974**). *Biochemistry, 13*, 222-245.

[63] Levin, J. M., Robson, B., and Garnier, J. (**1986**). *FEBS Lett., 205*, 303-308.

[64] Qian, N. and Sejnowski, T. J. (**1988**). *J. Mol. Biol., 202*, 865-884.

[65] Zvelebil, M. J., Barton, G. J., Taylor, W. R., and Sternberg, M. J. (**1987**). *J. Mol. Biol., 195*, 957-961.

[66] Stultz, C. M., White, J. V., and Smith, T. F. (**1993**). *Protein Science, 2*, 305-314.

[67] Bystro,, C. and Baker, D. (**1998**). *J. Mol. Biol., 281*, 565-577.

[68] Hirokawa, T., Boon-Chieng, S., and Mitaku, S. (**1998**). *Bioinformatics, 14*, 378-379.

[69] Guermeur, Y., Geourjon, C., Gallinari, P., and Deleage, G. (**1999**). *Bioinformatics, 15*, 413-421.

[70] Geourjon, C. and Deleage, G. (**1994**). *Protein Engineering, 7*, 157.

[71] King, R. D. and Sternberg, M. J. (**1996**). *Protein Science, 5*, 2298-2310.

[72] Frishman, D. and Argos, P. (**1996**). *Protein Eng., 9*, 133-142.

[73] Eisenhaber, F., Imperiale, F., Argos, P., and Froemmel, C. (**1996**). *Proteins, Struct. Funct. Genet., 25*, 157-168.

[74] Eisenhaber, F., Imperiale, F., and Argos, P. (**1996**). *Proteins, Struct. Funct. Genet., 25*, 169-179.

[75] Tripos Associates (**1999**). *SYBYL* 6.5.3. Tripos Associates, Inc., St. Lious, Missouri.

[76] Group, M. A. (**1999**). *LOOK version 3.5.1. Molecular Applications Group*, Palo Alto, Carlifornia.

[77] Humphrey, W. F., Dalke, A., and Schulten, K. (**1996**). *J. Mol. Graphics, 14*, 33-38.

[78] Sayle, R. A. and Milner-White, E. J. (**1995**). *Trends in Biochemical Sciences, 20*, 374-376.

[79] Kraulis, P. (**1991**). *J. Appl. Cryst., 24*, 946-950.

[80] MDL Information Systems (**1999**). *CHIME. MDL Information Systems*, Inc., San Leandro, Carlifornia.

[81] Flores, T. P., Moss, D. S., and Thornton, J. M. (**1994**). *Protein Eng., 7*, 31-37.

[82] Nicholls, A., Sharp, K. A., and Honig, B. (**1991**). *Proteins, Structure, Function and Genetics, 11*(4), 281-296.

[83] Kabsch, W. and Sander, C. (**1983**). *Biopolymers, 22*, 2577-2637.

[84] McDonald, I. K. and Thornton, J. M. (**1994**). *J. Mol. Biol., 238*, 777-793.

[85] Lee, B. and Richards, F. M. (**1971**). *J. Mol. Biol., 55*, 379-400.

[86] Richards, F. M. (**1977**). *Ann. Rev. Biochem. Bioeng., 6*, 151-176.

[87] Hubbard, S. and Thornton, J. (**1996**). *NACCESS*. EMBL, U.K.

[88] Eisenhaber, F. and Argos, P. (**1995**). *J. Comp. Chem., 16*, 273-284.

[89] Connolly, M. L. (**1993**). *J. Mol. Graphics, 11*, 139-141.

[90] Xie, D. and Freire, E. (**1994**). *J. Mol. Biol., 242*, 62-80.

[91] Xu, D. and Nussinov, R. (**1997**). *Fold. & Des., 3*, 11-17.

[92] King, R. D. and Sternberg, M. J. (**1995**). *Protein Eng., 8*, 513-525.

[93] Holm, L. and Sander, C. (**1998**). *Proteins, Struct. Funct. Genet., 33*, 88-96.

[94] Siddiqui, A. S. and Barton, G. J. (**1995**). *Protein Science, 4*, 872-884.

[95] Bernstein, F. C., Koetzle, T. F., Williams, G. J. B., Meyer, E. F., Brice, M. D., Rodgers, J. R., Kennard, O., Shimanouchi, T., and Tasumi, M. (**1977**). *J. Mol. Biol., 112*, 535-542.

[96] Laskowski, R. A., MacArthur, M. W., Moss, D. S., and Thornton, J. M. (**1993**). *J. Appl. Cryst., 26*, 283-291.

[97] Pontius, J., Richelle, J., and Wodak, S. J. (**1996**). *J. Mol. Biol., 264*, 121-136.

[98] Gibrat, J. F., Madej, T., and Bryant, S. H. (**1996**). *Curr. Opinion Struct. Biol., 6*, 377-385.

[99] Alexandrov, N. N. (**1996**). *Protein Eng., 9*, 727-732.

[100] Feng, Z. K. and Sippl, M. J. (**1996**). *Fold. & Des., 1*, 123-132.

[101] Holm, L. and Sander, C. (**1993**). *J. Mol. Biol., 233*, 123-138.

[102] Murzin, A. G., Brenner, S. E., Hubbard, T., and Chothia, C. (**1995**). *J. Mol. Biol., 247*, 536-540.

[103] Orengo, C. A., Michie, A. D., Jones, S., Jones, D. T., Swindells, M. B., and Thornton, J. M. (**1997**). *Structure, 5*, 1093-1108.

[104] Shindyalov, I. N. and Bourne, P. E. (**1998**). *Protein Eng., 11*, 739-747.

[105] Holm, L. and Sander, C. (**1996**). *Science, 273*, 595-602.

[106] Karplus, M. and McCammon, J. A. (**1983**). *Ann. Rev. Biochem., 53*, 263-300.

[107] McCammon, J. A. and Harvey, S. C. (**1987**),*Dynamics of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge.

[108] Levitt, M. and Warshel, A. (**1975**). *Nature, 253*, 694-698.

[109] McCammon, J. A., Gelin, B. R., and Karplus, M. (**1977**). *Nature, 267*, 585-590.

[110] Gunsteren, W. F. v. and Berendsen, H. J. C. (**1977**). *Mol. Phys., 34*(5), 1311-1327.

[111] Brooks, B. R., Bruccoleri, R. E., Olafson, B. D., States, D. J., Swaminathan, S., and Karplus, M. (**1983**). *J. Comp. Chem., 4*, 187-217.

[112] Gunsteren, W. F. v. and Berendsen, H. J. C. (**1987**). *GROMOS Manual. BIOMOS b. v.*, Lab. of Phys. Chem., Univ. of Groningen.

[113] Weiner, S., Kollman, P., Case, D., Singh, U., Ghio, C., Alagona, G., Profeta, J., and P.Weiner (**1984**). *J. Appl. Cryst., 106*, 765-784.

[114] Ponder, J. W. and Richards, F. M. (**1987**). *J. Comp. Chem.*, *8*, 1016-1024.

[115] Br-unger, A. T. (**1992**). *X-PLOR, Version 3.1, A System for X-ray Crystallography and NMR*. The Howard Hughes Medical Institute and Department of Molecular Biophysics and Biochemistry, Yale University, New Haven.

[116] Nelson, M., Humphrey, W., Gursoy, A., Dalke, A., Kale, L., Skeel, R. D., and Schulten, K. (**1996**). *Int. J. of Supercomputer Applications and High Performance Computing*, *10*, 251-268.

[117] Watanabe, M. and Karplus, M. (**1995**). *J. Phys. Chem.*, *99*(15), 5680-5697.

[118] Balsera, M. A., Wriggers, W., Oono, Y., and Schulten, K. (**1996**). *J. Phys. Chem.*, *100*(7), 2567-2572.

[119] Brooks, B. R. and Hodo-s-cek, M. (**1992**). *Chemical Design* Automation News (CDA News), *7*, 16-22.

[120] Kale, L., Skeel, R., Bhandarkar, M., Brunner, R., Gursoy, A., Krawetz, N., Phillips, J., Shinozaki, A., Varadarajan, K., and Schulten, K. (**1999**). *J. Comp. Phys.*, *151*, 283-312.

[121] Xu, D., Sheves, M., and Schulten, K. (**1995**). *Biophys. J.*, *69*(6), 2745-2760.

[122] Lu, H. and Schulten, K. (**1999**). *Proteins, Struct. Funct. Genet.*, *35*, 453-463.

[123] Gaussian, Inc. (**1998**). *Gaussian 98. Gaussian,* Inc., Pittsburgh, Pennsylvania.

[124] Schmidt, M. W., Baldridge, K. K., Boatz, J. A., Elbert, S. T., Gordon, M. S., Jensen, J. H., Koseki, S., Matsunaga, N., Nguyen, K. A., Su, S., Windus, T. L., Dupuis, M., and Montgomery, J. A. (**1993**). *J. Comp. Chem.*, *14*, 1347-1363.

[125] Q-Chem, Inc. (**1998**). *Q-Chem 1.2. Q-Chem*, Inc., Pittsburgh, Pennsylvania.

[126] Singh, U. C. and Kollman, P. A. (**1986**). *J. Comp. Chem.*, *7*, 718.

[127] Field, M. J., Bash, P. A., and Karplus, M. (**1990**). *J. Comp. Chem.*, *11*(6), 700-733.

[128] Aquist, J. and Warshel, A. (**1993**). *Chem. Rev.*, *93*, 2523-2544.

[129] Gilson, M. K., Rashin, A., Fine, R., and Honig, B. (**1985**). *J. Mol. Biol.*, *183*, 503-516.

[130] Juffer, A. H., Botta, E. F. F., Keulen, B. A. M. v., Ploeg, A. v. d., and Berendsen, H. J. C. (**1991**). *J. Comp. Phys.*, *97*(1), 144-171.

[131] Honig, B. and Nicholls, A. (**1995**). *Science, 268*, 1144-1149.

[132] Nilges, M. and Br-unger, A. (**1993**). *Proteins, Struct. Funct. Genet.*, *15*, 133-146.

[133] Hu, X., Xu, D., Hamer, K., Schulten, K., Koepke, J., and Michel, H. (**1995**). *Protein Science, 4*, 1670-1682.

[134] Madej, T., Gibrat, J. F., and Bryant, S. H. (**1995**). *FEBS Lett.*, *373*, 13-18.

[135] CASP (**1995**). *Proteins, Struct. Funct. Genet.*, *23*, 295-462.

[136] CASP (**1997**). Proteins, Struct. Funct. *Genet.*, Suppl. 1, *29*, 1-230.

[137] CASP (**1999**). Proteins, Struct. Funct. *Genet.*, Suppl. 3, *37*, 1-237.

[138] Bower, M., Cohen, F., and Dunbrack Jr, P. L. (**1997**). *J. Mol. Biol.*, *267*, 1268-1282.

[139] Holm, L. and Sander, C. (**1991**). *J. Mol. Biol.*, *218*, 183-194.

[140] Peitsch, M. C. (**1996**). *Biochem. Soc. Trans.*, *24*, 274-279.

[141] Lund, O., Frimand, K., Gorodkin, J., Bohr, H., Bohr, J., Hansen, J., and Brunak, S. (**1997**). *Protein Eng.*, *10*, 1241-1248.

[142] Srinivasan, B. N. and Blundell, T. L. (**1993**). *Protein Eng.*, *6*, 501-512.

[143] Sali, A. and Blundell, T. L. (**1993**). *J. Mol. Biol.*, *234*, 779-815.

[144] Bowie, J. U., Luthy, R., and Eisenberg, D. (**1991**). *Science, 253*, 164-170.

[145] Sippl, M. J. and Weitckus, S. (**1992**). *Proteins, Struct. Funct. Genet.*, *13*, 258-271.

[146] Jones, D. T., Taylor, W. R., and Thornton, J. M. (**1992**). *Nature, 358*, 86-89.

[147] Xu, Y., Xu, D., and Uberbacher, E. C. (**1998**). *J. Comp Biol.*, *5*(3), 597-614.

[148] Crawford, O. H. (**1999**). *Bioinformatics, 15*, 66-71.

[149] Smith, T., Conte, L. L., Bienkowska, J., Gaitatzes, C., Rogers, R., and Lathrop, R. (**1997**). *J. Comp Biol.*, *4*(3), 217-225.

[150] Alexandrov, N. N., Nussinov, R., and Zimmer, R. M. (**1996**). *In Hunter, L. and Klein, T., Eds., Biocomputing*, Proceedings of the 1996 Pacific Symposium, pages 53-72. World Scientific Publishing Co., Singapore.

[151] Rost, B. (**1995**). *ISMB, 3*, 314-321.

[152] Milburn, D., Laskowski, R. A., and Thornton, J. M. (**1998**). *Protein Eng., 11*, 855-859.

[153] Fischer, D. and Eisenberg, D. (**1996**). *Protein Science, 5*, 947-955.

[154] Bryant, S. H. and Lawrence, C. E. (**1993**). *Proteins, Struct. Funct. Genet.*, *16*, 92-112.

[155] Li, Z. and Scheraga, H. A. (**1987**). *Proc. Natl. Acad. Sci. USA, 84*, 6611-6615.

[156] Friedrichs, M. S. and Wolynes, P. G. (**1989**). *Science, 246*, 371.

[157] Skolnick, J. and Kolinski, A. (**1991**). *J. Mol. Biol., 221*, 499-531.

[158] Sali, A., Shakhnovich, E., and Karplus, M. (**1994**). *J. Mol. Biol., 235*, 1614-1636.

[159] Pedersen, J. T. and Moult, J. (**1997**). *J. Mol. Biol., 269*, 240-259.

[160] Unger, R. and Moult, J. (**1992**). *J. Mol. Biol., 5*, 637-645.

[161] Srinivasan, R. and Rose, G. (**1995**). *Proteins, Struct. Funct. Genet., 22*, 81-99.

[162] Simons, K. T., Kooperberg, C., Huang, E., and Baker, D. (**1997**). *J. Mol. Biol., 268*, 209-225.

[163] Kuntz, I. D., Blaney, J. M., Oatley, S. J., Langridge, R., and Ferrin, T. E. (**1982**). *J. Mol. Biol., 161*, 269-288.

[164] Fischer, D., Lin, S. L., Wolfson, H. J., and Nussinov, R. (**1995**). *J. Mol. Biol., 248*, 459-477.

[165] Vakser, I. A. and Aflalo, C. (**1994**). *Proteins, Struct. Funct. Genet., 20*, 320-329.

[166] Wallqvist, A. and Covell, D. G. (**1996**). *Proteins, Struct. Funct. Genet., 25*, 403-419.

[167] Morris, G. M., Goodsell, D. S., S., H. R., Huey, R., Hart, W. E., Belew, R. K., and Olson, A. J. (**1998**). *J. Comp. Chem., 19*, 1639-1662.

[168] Subramaniam, S. (**1998**). *Proteins, Struct. Funct. Genet., 32*, 1-2.

[169] Gerstein, M. (**1997**). *J. Mol. Biol., 274*, 562-576.

[170] Fischer, D. and Eisenberg, D. (**1997**). *Proc. Natl. Acad. Sci. USA, 94*, 11929-11934.

[171] Jones, D. T. (**1999**). *J. Mol. Biol., 287*, 797-815.

[172] Ding, H.-Q., Karasawa, N., and Goddard III, W. A. (**1991**). *Bulletin of the American Physical Society, 36*(6).

[173] Wriggers, W., Mehler, E., Pitici, F., Weinstein, H., and Schulten, K. (**1998**). *Biophys. J., 74*, 1622-1639.

[174] Schlick, T., Skeel, R., Br-unger, A., Kale, L., Board Jr., J. A., Hermans, J., and Schulten, K. (**1999**). *J. Comp. Phys., 151*, 9-48.

[175] Ferrin, T. E., Couch, G. S., Huang, C. C., Pettersen, E. F., and Langridge, R. (**1991**). *J. Mol. Graphics, 9*, 27-32.

[176] Surles, M. C., Richardson, J. S., Richardson, D. C., and Brooks, F. P. (**1994**). *Protein Science, 3*, 198-210.

[177] Dalke, A. and Schulten, K. (**1997**). In Proceedings of the Pacific Symposium on Biocomputing 97 on Interactive Molecular Visualization, pages 85-96,.

[178] Lin, S. L., Xu, D., Li, A., Roiterst, M., Wolfson, H. J., and Nussinov, R. (**1997**). *J. Mol. Biol., 271*.

[179] Lin, S. L., Xu, D., Li, A., Roiterst, M., Wolfson, H. J., and Nussinov, R. (**1998**). *Proteins, Struct. Funct. Genet., 31*.

[180] Xu, Y., Xu, D., Crawford, O. H., Einstein, J. R., Larimer, F., Uberbacher, E. C., Unseren, M. A., and Zhang, G. (**1999**). *Protein Eng., 12*, 899-907.