A NOVEL FRAMEWORK FOR PROTEIN STRUCTURE PREDICTION

A Dissertation presented to the
Faculty of the Graduate School
University of Missouri-Columbia

In partial fulfillment of the requirements for the Degree
Doctor of Philosophy

By
RAJKUMAR BONDUGULA

Dong Xu, PhD
Dissertation Supervisor

May 2007

The undersigned, appointed by the Dean of the Graduate School, have examined the dissertation entitled

A NOVEL FRAMEWORK

FOR PROTEIN STRUCTURE PREDICTION

presented by Rajkumar Bondugula,

a candidate for the degree of Doctor of Philosophy,

and hereby certify that, in their opinion, it is worthy of acceptance.

Professor Dong Xu

Professor Ioan kosztin

Professor James Keller

Professor Chi-Ren Shyu

Professor Yi Shang

Dedicated to:

My parents, who gave me life.

My teachers, who taught me the essence of life.

ACKNOWLEDGEMENTS

I would like to express my gratitude to Dong Xu, Ph.D. for giving me an opportunity to work with him. I would also like to thank him for his expert tutelage, patience, guidance, and confidence entrusted upon me to complete the research in this topic area, without which this project would not have been successful.

I acknowledge James Keller, Ph.D. for his support. His patience, guidance and insight have been invaluable assets throughout the course of my study.

Yi Shang, Ph.D., and Ioan Kosztin, Ph.D., have been great source of inspiration and encouragement to me throughout our collaborative effort. I would like to covey a special thanks to them.

Chi-Ren Shyu, Ph.D., made excellent suggestions that prompted me to think deeply and look at my project critically. For his support, I express my gratitude.

I would express my heartfelt thanks to my parents, my brother Pavan, my sister Swapna and other family members for their encouragement and support during my entire course of study. A special thanks to my uncle Hariprasad Gali, Ph.D., without whose inspiration, I would have never come into research.

I would like to thank my colleagues Tran Hong Nha Nguyen, Jingfen Zhang, Ph.D., Gyanprakash Srivastava, Bogdan Barz, Qingguo Wang, my former colleagues Mihail Popescu, Ph.D., Ozy Sjahputera, Ph.D., Samer Arafat, Ph.D., and others for their support throughout my research.

Finally, I would like to convey a special thanks my friends Dhanarekha Vasireddy, Megan Makarewicz, Stephanie Henderson and Anoop Haridas for their support and constant encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

1.1 Motivation

Proteins are one of the most important molecules in life. They play a variety of roles depending on their types including structural proteins, catalytic proteins, storage and transport proteins, regulatory proteins, immune system proteins, signaling proteins, and so on. A protein is a sequence of amino acids that are linked by peptide bonds to form a poly-peptide chain called its primary structure. Short runs of these amino acids form regular structures called secondary structures. There are three types of secondary structures, i.e., Helices, Strands, and Coils. The secondary structure elements are packed together into compact tertiary structure of the protein. Sometimes multiple tertiary structure elements from different poly-peptide chains are packed into a complex called quaternary structure. Various levels of protein structures are illustrated in Figure 1.1.

The shape and the function of a protein are related. For example, the structure of hemoglobin with four poly-peptide chains allows it bind and transport oxygen; collagen in its triple helix confirmation has high tensile strength, making it suitable for connecting tissue; insulin fits in spaces like a key in a keyhole, by which it controls the sugar levels. Therefore, the structure of a protein is essential for understanding its function at the molecular level. Anfinsen [Anfinsen *et al.*, 1961] discovered that all information required for a protein to fold in to a unique confirmation, in a given environment, is fully encoded in its sequence. Theoretically, a given protein sequence with 100 amino acids could fold into any of its 100^{20} (there are 20 naturally occurring amino acids) possible

conformations. However, only a few hundred to few thousand folds are observed in nature [Levinthal, 1968]. Lot of research [Li and Sheraga, 1987; Bowie and Eisenberg, 1991; Ring and Cohen, 1993; Simmons et al, 1997; Bystroff and Baker, 1999; Inbar et al, 2003; Chikenji et al, 2003; Lee et al, 2004; Ginalski et al, 2005; Moult 2005; Moult, 2006; Baker, 2006] was put into understanding the sequence-structure relationship.

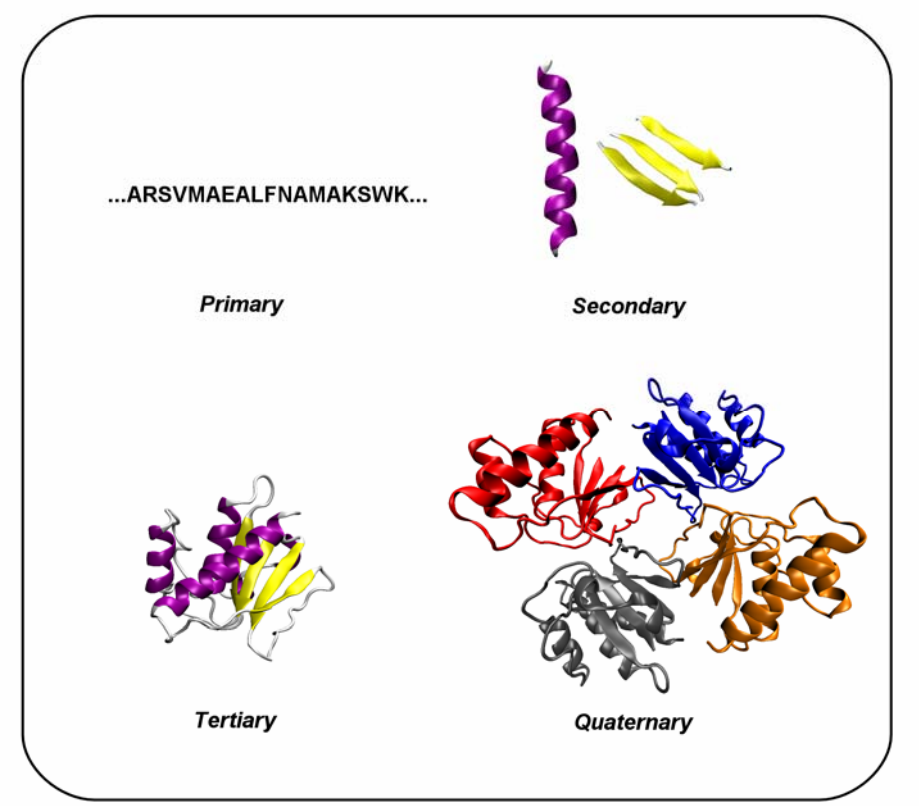


Figure 1.1: Protein structure hierarchy. The illustrated protein is an arsenate reductase from the species *Archaeoglobus* (PDB code 1Y1L).

The ability to determine/predict the structures of proteins from their amino acid sequences is of central importance to contemporary molecular biology. Traditionally, protein structures are solved using the X-ray crystallography or NMR methods. It can

take months or even years to solve one structure, which can cost hundreds of thousands of dollars. An alternative approach for protein structure solution is through computational prediction. The key advantages of computational methods are their speed and cost – the solution time can be reduced to hours or even minutes with little cost. Although current prediction methods can often provide useful structural information, they are unable to consistently produce structures of comparable quality to those produced by experimental methods. Protein structure prediction remains one of the most important and challenging problems in computational biology [Ginalski *et al.*, 2005; Moult, 2005; Aloy *et al.*, 2005; Dunbrack, 2006; Ginalski, 2006].

Research in protein structure prediction is especially timely because of rapid advancements being made in the Human Genome Project and other genome sequencing projects. These efforts are devoted to sequence DNA fragments, i.e., to determine the order of nucleic acids therein. The sequence largely consists of a set of blueprints for proteins. Once the sequence of a gene is known, the amino acid sequence for the protein coded by that gene can be annotated. Various genome projects have produced millions of new proteins, whose structures are unknown experimentally but could be predicted computationally. The ability to predict protein structure has a proven impact on pharmaceutical and biotechnological research. The 3D structure of protein holds the key in understanding its biological function at the molecular level. Knowledge of protein structure also allows researchers to identify and characterize disease targets and provides a rational approach to drug design. For example, effective drugs were derived to target the AIDS virus based on the structure of HIV protease [Kitchen et al., 2004].

1.2 Novel Framework for Protein Structure Prediction

In this dissertation, we present a novel framework we developed that can be used to predict many aspects of protein structure. Our framework is a suite of programs that perform general tasks, whose output will be used by other programs to predict specific aspects of protein structures. We have used the framework successfully in three applications, two of which have the state-of-the-art performance, i.e., protein secondary structure prediction and protein solvent accessibility prediction. The other application with great potential and some promising results is protein tertiary structure prediction. We expect that the same frame-work can be used for contact map prediction.

The role of our framework can be briefly explained in the following phrase "fast and efficient remote compatible fragment finder and assembler for a given sequence to infer structural information". Like the phrase suggests, given a query protein sequence, our application retrieves compatible fragments that are similar to the sub-sequences of the query protein from the database of proteins, whose structures were determined experimentally. This framework will enable the applications to avoid time consuming systematic search in the database for homologous fragments. It relies on high-speed heuristic based algorithms to retrieve the compatible fragments from large sequence databases. The applications use these compatible fragments to predict specific structural aspects of proteins. These fragments will be used to predict the features of the protein for which there are no sequence level homologs in the database of known protein structures. The framework has parsers to fetch all the available structural information for the compatible fragments. The structural information will allow the applications to use as

much information as possible from the database of protein structures. The framework also retrieves the statistical significance of the sequence alignment that produced the compatible fragment. In addition to the compatible fragments, the framework also has parsers to retrieve the sequence profile information of the query sequence. The sequence profile captures the evolutionary history of the sequence and this information was proved to boost to performance of many bioinformatics applications, including the structural applications.

All the programs that are designed to use our framework have some application-specific components like the fundamental algorithm of the specific application, protein sequence databases used, scoring schemes and the benchmark datasets for comparing the application's performance with existing methods.

1.3 Organization

In Chapter 2, we explain our framework in detail. Specifically, we discuss the need for such a framework, the analysis that lead to discovery of optimal parameters, the components of the framework and the information produced by our framework. In Chapter 3, we explain the application of our framework for protein secondary structure prediction. We explain the fundamental concepts of protein secondary structure, the need for a method, our algorithm, results that include the performance comparison with existing methods. In Chapter 4, we describe application of our framework for protein solvent accessibility prediction. We elucidate the deficiencies in the existing method and how our method overcomes such deficiencies. In Chapter 5, we introduce a novel tertiary structure prediction method, based on our framework. We explain the novel formulation

of our method for protein tertiary structure prediction and explain how our method overcomes most of the problems in the existing methods. In Chapter 6, the dissertation is concluded and directions for possible future work are suggested.

2. A Novel Framework for Protein Structure Prediction

2.1 Motivation

For proteins that have close sequence level homologs in the database of proteins with known structures such as Protein Data Bank (PDB) [Berman *et al*., 2000], almost all aspects of the structure can be predicted very reliably, including the secondary and tertiary structure. On the other hand, most proteins do not share significant sequence similarity with any other sequences that have their three-dimensional structures determined experimentally. Predicting structural aspects of such proteins is a challenging and unsolved problem [Ginalski *et al.*, 2005; Moult, 2005; Aloy *et al.*, 2005].

Specifically, many existing methods for protein secondary structure and solvent accessibility prediction rely on the position specific scoring matrix (PSSM) [Altschul *et al.*, 1997] alone (more details on PSSM below in this chapter). For the sequences that have homologs in the database of known sequences such as *nr* (a large nucleotide database available through NCBI at *http://www.ncbi.nlm.nih.gov*. The name *nr* is derived from "non-redundant", but this is historical only, because this database is no longer non-redundant), the profile is well-defined. Otherwise, the profile is not well-defined and the predictions are unreliable [Geourjon and Deleage, 1994; Salamov and Solovyev, 1995; Adamczak *et al.*, 2004]. Relying on profile alone means inefficient use of the structural information in the PDB. These methods predict the structural features of proteins without using available structural information. We want to answer the following: can the methods

that predict the local structural features of the proteins take advantage of the structural information in conjuction with the sequence profile information? If yes, what kind of structural features are important? If the structural information is incorporated into the prediction system, what is extent of improvement?

In case of tertiary structure prediction, on one extreme, *ab initio* [Li and Sheraga, 1987; Pedersen and Moult, 1997] methods predict the structure of a protein from first principles, using neither structural nor profile information. *Ab initio* methods demand huge computing power and the results are generally unreliable. On the other extreme, homology modeling methods [Bowie and Eisenberg, 1991; Ring and Cohen 1993; Chivian and Baker, 2006] rely on close homologs. In absence of close homologs, these methods fail. Mini-threading [Simmons et al, 1997; Bystroff and Baker, 1999; Inbar et al, 2003; Chikenji et al, 2003; Lee et al, 2004] methods are a compromise between these two methods, where the structure of the query protein is predicted by assembling similar fragments. Existing methods that use this approach rely on slow systematic search of the fragments in the database. Is there a computationally efficient and faster way to search for similar fragments than systematic database search? Can we use the available structural information better than the existing methods?

In the light of above questions, we propose a framework that can do the following tasks: For a given query protein sequence,

1) quickly search for similar fragments and fetch all the available structural information of these fragments;

2) generate profile information and parse it into easily usable information by applications;
3) perform above tasks using minimum computing resources (CPU and RAM);
4) help build the applications that bridge the gap between various methods that rely on only one type of information (either profile or structure).

2.2 BLAST programs

The problem of finding common subsequence of two strings is a challenging. This problem was effectively tackled by the dynamic programming approach. The generalized form the problem, finding a common subsequence (or approximate substring) of many strings, however, is a bigger challenge that dynamic programming cannot solve efficiently. Several programs were written to address the problem. The development of the Basic Local Alignment and Sequencing Tool (BLAST) [Altschul et al, 1990] and Position Iterated BLAST (PSI-BLAST) [Altschul et al, 1997] has revolutionized several areas in bioinformatics.

The BLAST program is widely used tool for searching protein and DNA databases for sequence similarities. It is a heuristic based, efficient algorithm that performs approximate sequence alignments to search for fragments in the database quickly. It emphasizes regions of local alignment to detect relationships among sequences which share only isolated regions of similarity. The sequence alignments of various lengths generated by BLAST are accompanied by various statistics that include sequence similarity, percentage of identical amino acids in the alignment region, alignment length, raw and scaled scores of alignment and finally the expectation value (E-value). E-value is

a statistical significance measure that indicates the number of different alignments with scores as good as or better than S that are expected to occur in a database search by chance. The lower the E-value, the significant the alignment.

PSI-BLAST is an iterative search using the BLAST algorithm. A profile is built after the initial search, which is then used in subsequent searches. The process may be repeated, if desired with new sequences found in each cycle used to refine the profile. Profile or PSSM of a protein represents the position-dependent amino acid distribution derived from the multiple sequence alignments. An amino acid at a particular position that is highly conserved receives a higher score than one that is less conserved at that position. A protein of length *l* has a PSSM of dimension *l*x20. The PSSM gives the log-odds score for finding a particular matching amino acid in a target sequence.

For a given query sequence, we first construct the profile of the query sequence using PSI-BLAST and the *nr* database. We then use the generated profile to perform a profile-sequence alignment to search for homologous fragments. The multiple sequence alignment derived profile incorporates evolutionary information into the search process. This information will make the search process more sensitive and return remote similar fragments.

2.2.1 Issues to consider

The BLAST package from NCBI (*http://www.ncbi.nlm.nih.gov*) is a large (approximately 90,000 lines) suite of programs with many options to fine tune the behavior of the program. Some of the issues that need to be addressed are:

1. What mutation matrix is the best for PSI-BLAST and BLAST, respectively?
2. How many iterations of PSI-BLAST are the best to produce the PSSM?
3. What is the best E-value threshold for PSI-BLAST to produce the PSSM?
4. Can we guarantee a good coverage of hits throughout the length of sequence?
5. How much information is there in these hits? Can we estimate the secondary structure identity in the alignment region? Can we estimate the mean absolute error of the solvent accessibility in the alignment region? How similar are tertiary structures in the alignment region?

2.2.2 Large-scale experiments to discover parameters

In order to address the issues discussed above, Tran HN Nguyen, a graduate student in our laboratory has helped us in conducting large-scale experiments to discover the optimal parameters and to estimate the upper bounds of the information contained in the similar fragments. We need the following components for our experiments: BLAST suite of programs, a large database of sequences to build a profile and a database of proteins with known structures.

We experimented with BLAST suite versions 2.2.10, 2.2.11 and 2.2.12 (*ftp://ftp.ncbi.nlm.nih.gov/blast/executables*). We used the *nr* database for building the profile. We need a representative protein set (RPS) to search for the similar fragments. For this, we used the March 2006 release of *PDBSelect* [Hobohm and Sander, 1994] database. This database consists of representative proteins such that the sequence identity between any two proteins in the database is not more than 25%. Initially, the database has 3080 chains. This database was filtered to select high-quality structures. In particular,

only structures that are generated using the X-ray crystallography method with a resolution of 3 Å or less were selected. Of these, proteins with incomplete backbone atoms were discarded. Proteins that are shorter than 40 residues were also removed. Furthermore, if less than 90% of the protein residues are composed of regular amino acids, they are discarded too. Finally, the remaining 1998 proteins after the filtering process constitute our RPS. We use the following procedure to generate hits:

For i = 1 to 1998

1. Split the RPS in to two files. The first file contains the i^{th} sequence and is used as the query sequence. The second file contains the remaining 1998 sequences and is used as the database of representative proteins.
2. Generate a BLAST-compatible search database for file containing the representative proteins, using the '*formatdb*' program in the BLAST suite.
3. Build the profile of the query sequence using the PSI-BLAST program and the *nr* database.
4. Use the generated profile to perform profile-sequence alignment to search for similar fragments, using PSI-BLAST the second time.
5. Parse the output of the PSI-BLAST to collect the hits and statistics like alignment length, percentage sequence similarity, percentage sequence identity, raw scores and E-value.
6. For the hit fragments, get the following structural information: secondary structures, Φ/Ψ angles, solvent accessibility and Cartesian coordinates.

End For

The following parameters were employed to generate the profile: *j*(number of iterations to construct the profile) = 3, *e*(expectation value threshold) = 0.001, and M(scoring matrix) = BLOSUM90. We use the BLOSUM90 substitution matrix as we want only the hit fragments that are close to the subsequences of the query protein to contribute to the PSSM being generated. When the profile of the query protein is used to search for the similar fragments in RPS by running the PSI-BLAST the second time, the threshold value of *e* was set to 11,000 when searching the RPS. The selection of the threshold is based on two considerations. On one hand, lower threshold for the E-value will result in few, highly significant hits. On the other hand, a higher threshold leads to a larger number of hits that include both truly homologous fragments and noise from the database. Too much noise will lead to decreased prediction accuracy [Bondugula and Xu, 2007]. Further in the chapter, we will discuss the effect of varying these parameters. The structural information of the hit fragments is obtained from the DSSP [Kabsch and Sander, 1983] files of the protein in the RPS. The DSSP standard of eight secondary structures were reduced to the CASP standard of three-state secondary structures as follows: {H, G, I}→Helix, {E, B}→Strand, and {C, T, S}→Coil.

The above procedure resulted in 1,411,333 hit fragments (excluding the hits that are shorter than 3 residues in length). We analyzed these hits and generated various distributions. The distribution of hits of various lengths is presented in Figure 2.1. The average number of hits per protein is 680. The average length of hits is 18.26 residues with 14 residues as median hit length. Next, we analyzed the distribution of the statistical significance (E-values) of the hits. The distribution is presented in Figure 2.2. The hits

represented by the left most bars have high E-values, therefore low significance. The rightmost bars represent the most significant hits, which are highly informative. We now proceed to distributions of the structural information. First, we examine the distribution of hits with various secondary structure similarity percentages in the alignment regions. The percentage similarity is calculated using the following formula:

$$\text{secondary structure similarity} = \frac{\text{no. of residues with same secondary structure}}{\text{alignment length}} \times 100 . \quad (2.1)$$

It can be noticed that hits having almost all percentages of similarities can be found. We present the actual estimations further in the chapter. Second, we examine the dihedral angle deviation distribution. Figures 2.4 and 2.5 contain Φ angle and Ψ angle distributions, respectively. We used the following formulae to calculate the dihedral angle deviations:

$$\Delta\Phi = \frac{\sum_{1}^{alignment\ length} \left|\Phi_{query} - \Phi_{hit}\right|}{alignment\ length}, \quad (2.2)$$

$$\Delta\Psi = \frac{\sum_{1}^{alignment\ length} \left|\Psi_{query} - \Psi_{hit}\right|}{alignment\ length}. \quad (2.3)$$

It can be noticed from these distributions that most of hits have Φ/Ψ deviations of about 50 and also that the ϕ angle has a narrower distribution. Next, we move on to present the distribution of mean average error (MAE) of the relative solvent accessibility (RSA). The RSA can be calculated by dividing the absolute solvent accessibility returned by the DSSP with their maximum solvent accessibility. We use the maximum solvent accessibilities from [Rost and Sander, 1994]. The MAE is calculated as follows:

$$MAE = \frac{\sum_{1}^{alignment\ length} \left| RSA_{query} - RSA_{hit} \right|}{alignment\ length}. \qquad (2.4)$$

The distributions of MAE are presented in Figure 2.6. Finally, we present the distribution of root mean square deviation (RMSD) between the protein fragments in the alignment regions. The RMSD was calculated using the VMD program (*http://www.ks.uiuc.edu/Research/vmd*). The distribution is presented in Figure 2.7.

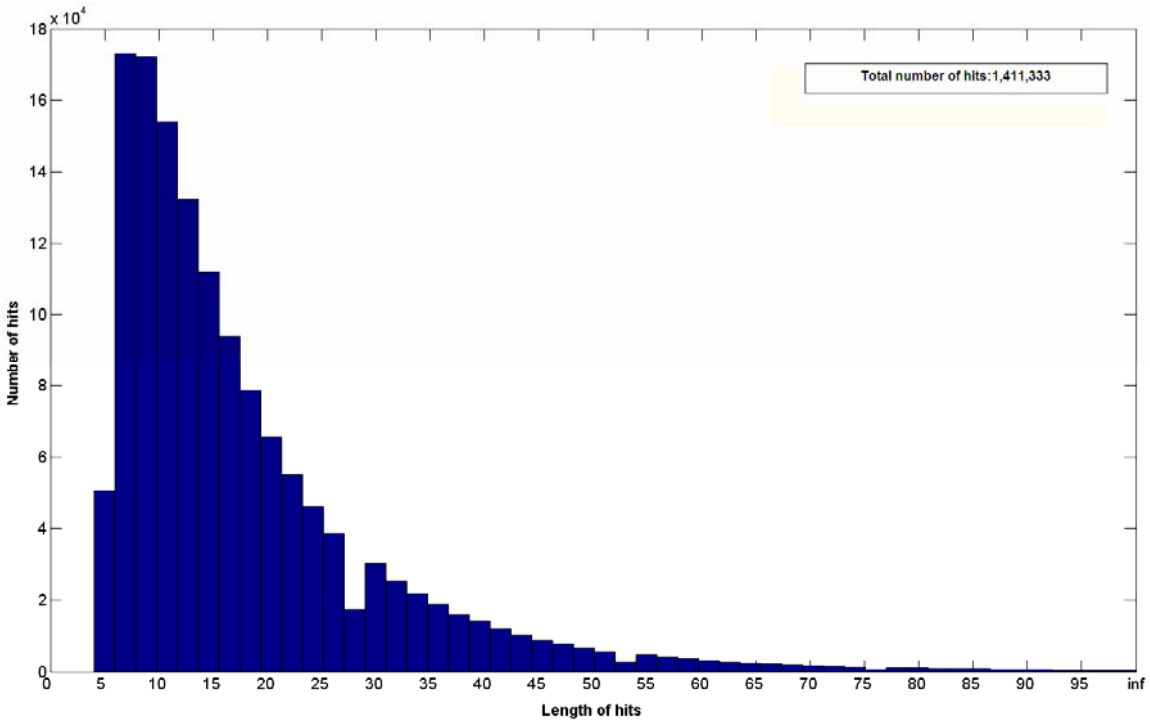


Figure 2.1: The distribution of hits of various lengths.

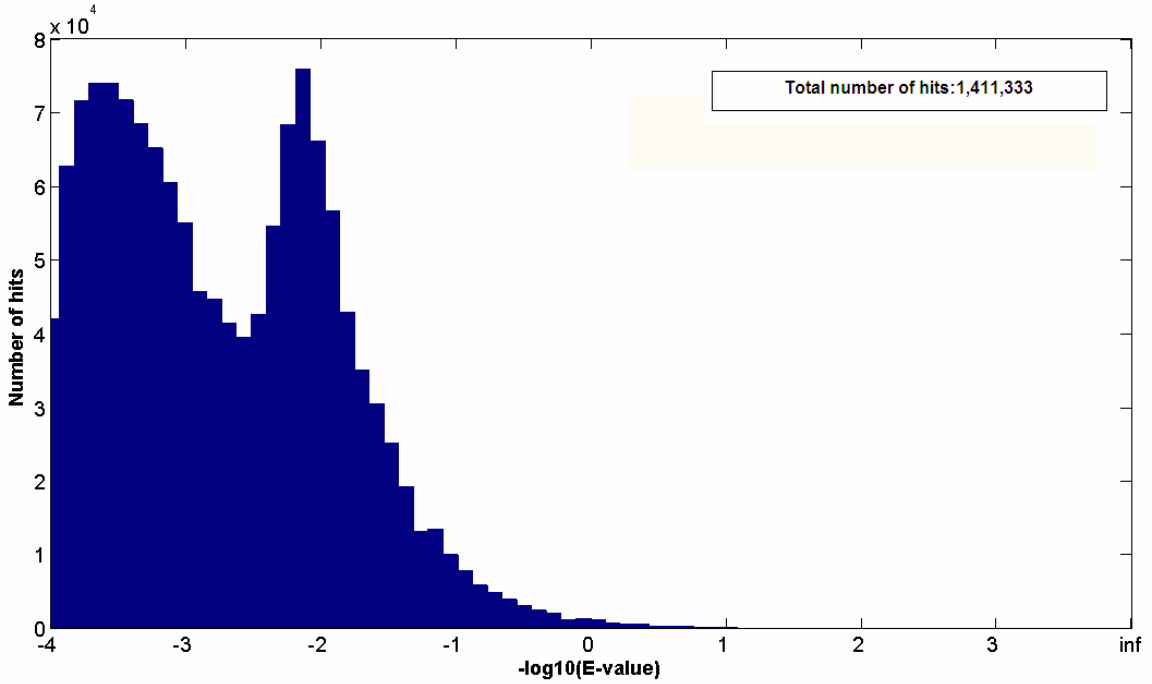


Figure 2.2: The distribution of E-values of the hits. The hits represented by the left most bars have high E-values, therefore low significance. The rightmost bars represent the most significant hits, which are highly informative.

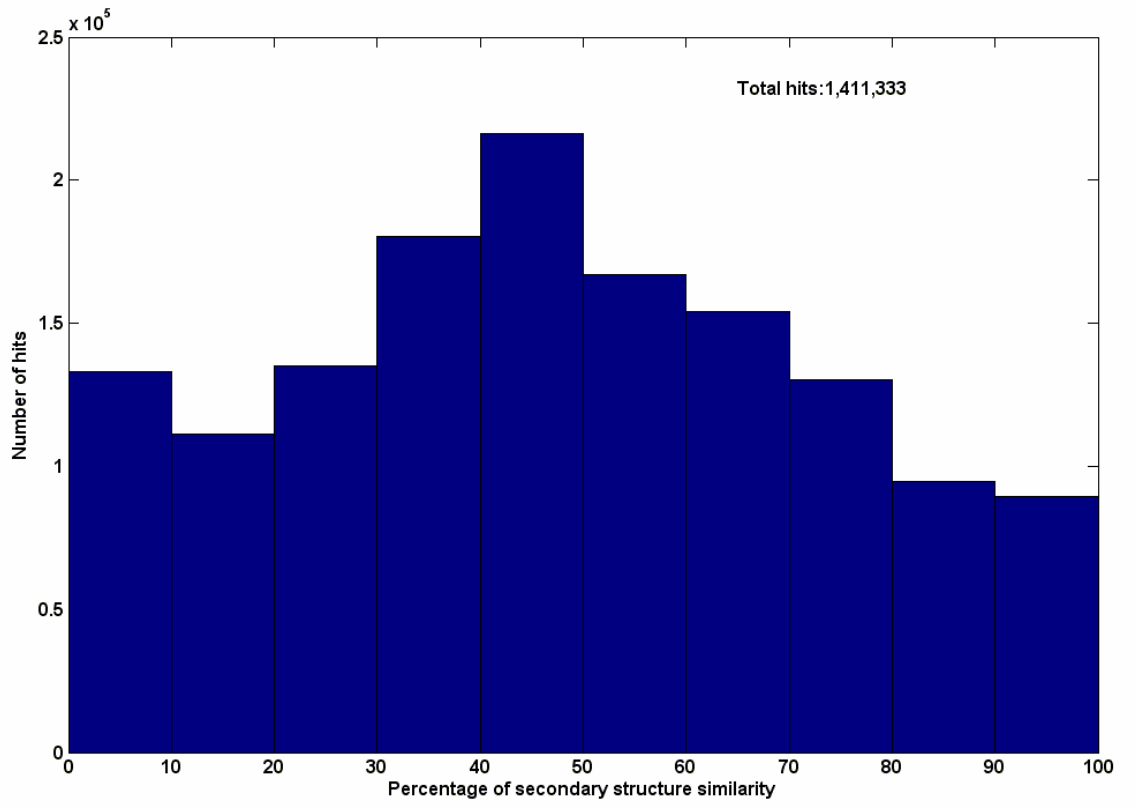


Figure 2.3: The distribution of hits with various secondary structure similarity percentages. The hits span all regions of similarity percentages.

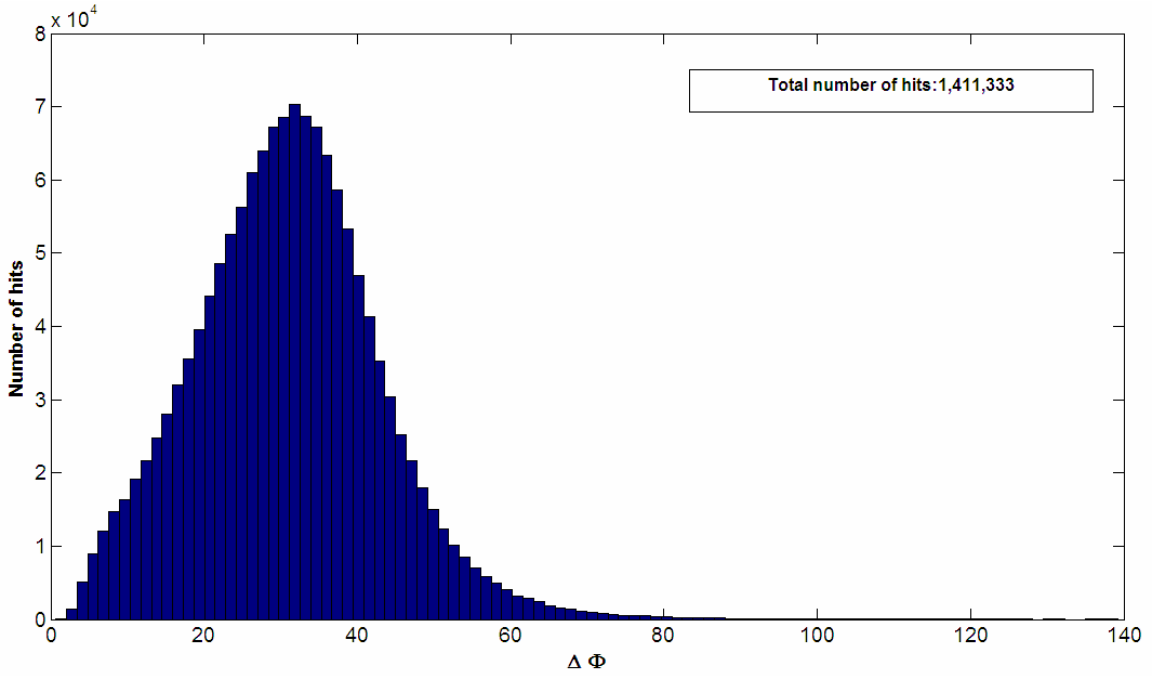


Figure 2.4: Distribution of Φ angle deviation across all hits.

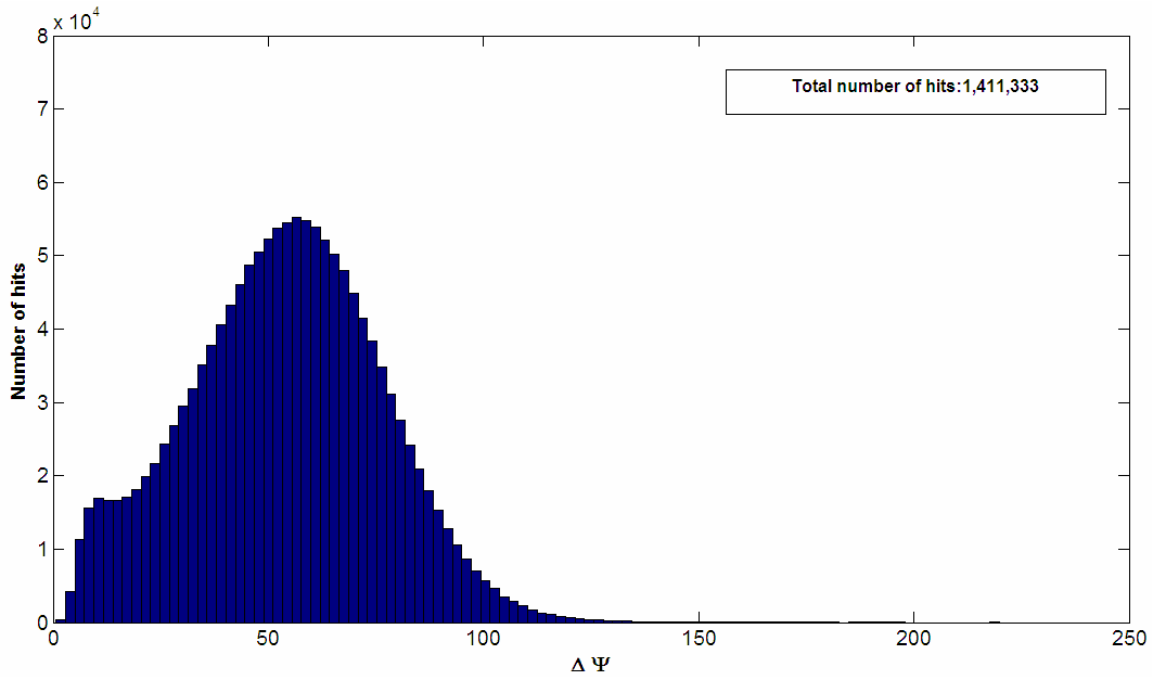


Figure 2.5: The distribution of Ψ angle deviation across all hits.

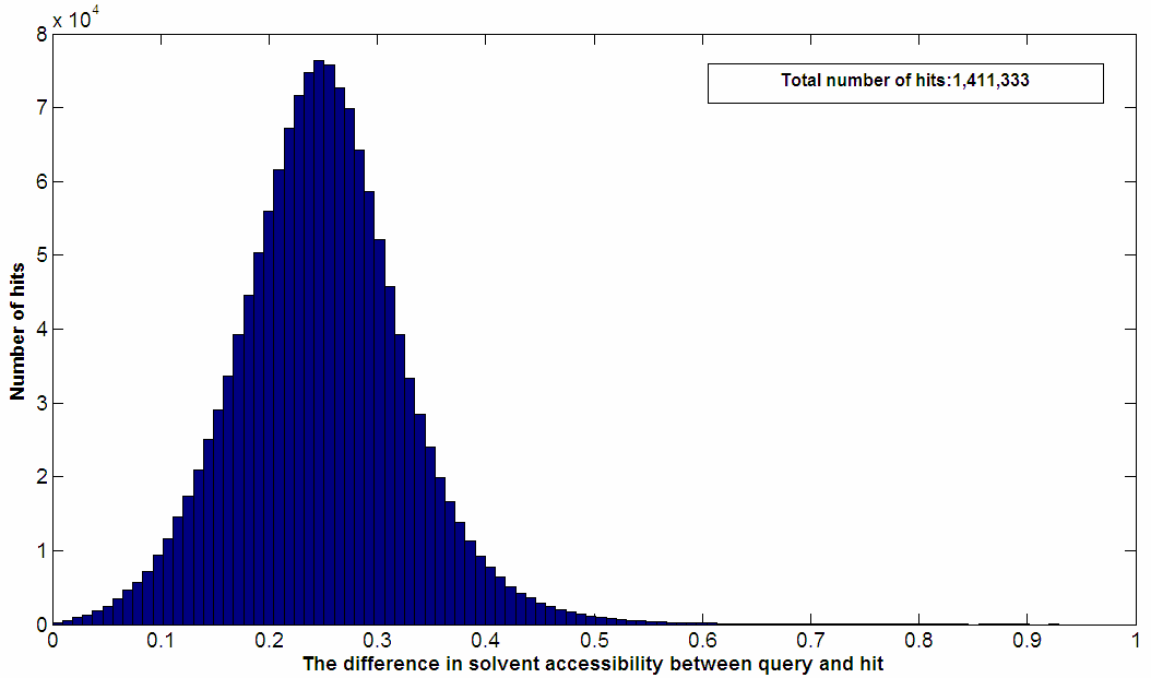


Figure 2.6: The distribution of mean absolute difference of the solvent accessibilities between the query fragments and the hits.

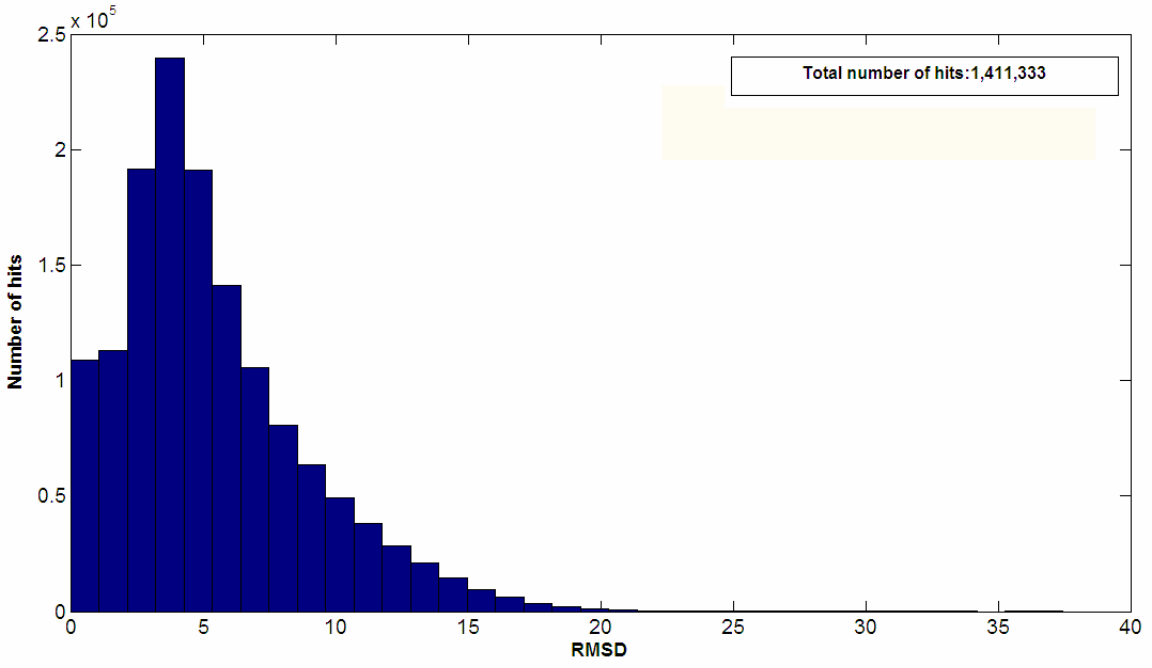


Figure 2.7: The distribution of RMSD values between the query fragments and the hits.

While the distributions presented so far give us estimates about some specific structural aspect, we now examine the number of alignments found in each position of the query sequence. The sequences in the RPS vary in length from 50 amino acids to more than 1,300 amino acids. In order to dissociate length as a variable, we present the results using normalized position. The plot is given in the Figure 2.8. From this plot, we can observe that uniform number of hits can be obtained for about 80% of the sequence length, with reduced number of hits in the termini of the protein.

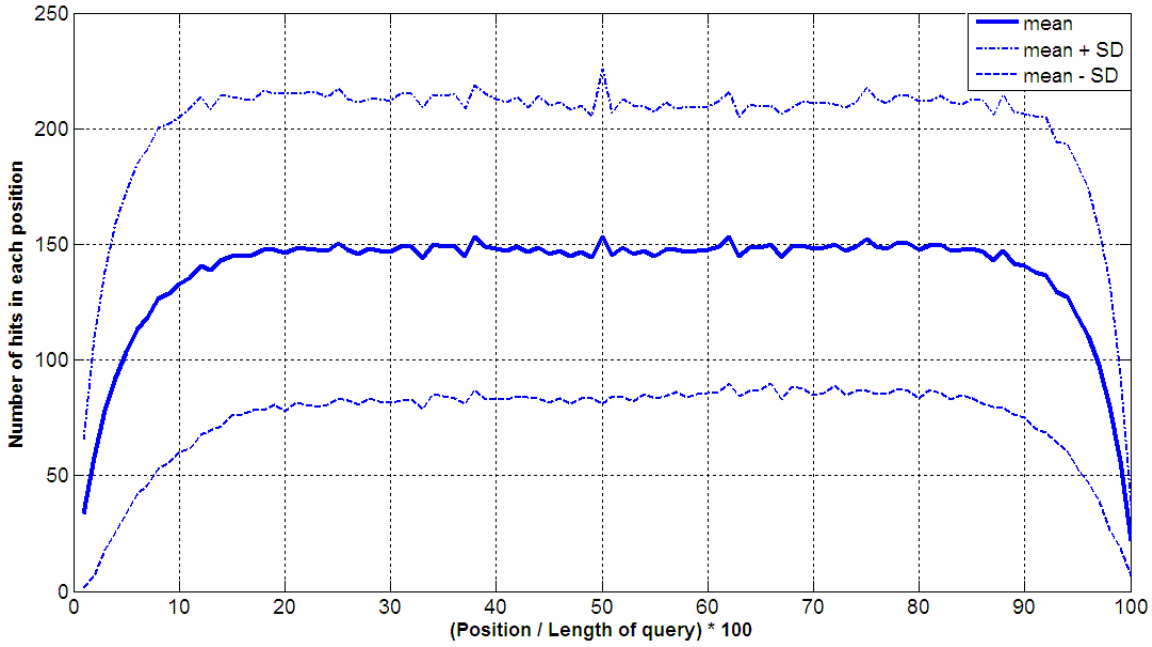


Figure 2.8: The plot depicts the number of alignments found in each position, normalized by the length of the protein. About 80% of the protein sequences have a uniform coverage of alignments. SD stands for standard deviation at each position.

We now present some plots that give us specific estimates of various structural features, for a hit with a given E-value. We also study the effect of various parameters for the PSI-

BLAST program. First, we present the plot (Figure 2.9) that gives us an estimated secondary structure similarity and MAE of the solvent accessibility of a given fragment with a certain E-value. The secondary structure similarity on the vertical axis on the left is plotted with negative logarithm of E-value on the horizontal axis. The mean absolute error of the solvent accessibility is plotted using the vertical axis on the right and the log-transformed E-value on the horizontal axis. On average, for hits on the higher end of statistical significance, the secondary structure similarity is 80% and the MAE of solvent accessibility is 0.18. These numbers are helpful in estimating the accuracies of the secondary structure and solvent accessibility predictions. We will see that the applications based on our framework reached or exceeded these estimates. Next, we present a plot that is useful to estimate the dihedral angle (Φ/Ψ) deviation, given a fragment with an E-value in Figure 2.10. It can be noticed that in general, the Φ angle has lower deviation than the Ψ angle. We now proceed to present the plot to estimate RMSD between the fragments in the alignment region. This plot is different from other feature plots, as the RMSD also depends on the alignment length (hits of same statistical significance have different RMSD for alignments of different lengths). For this purpose, we first found the minimum alignment length that shows correlation with the E-value. We found that alignments with length of 8 or more show clear correlation between RMSD and the E-value. We then divided all hits into bins of different lengths and plotted the relationship. The plots are depicted in Figure 2.11. This information is useful in protein tertiary structure prediction. These plots suggest us excluding fragments that are

too small to contribute usefully to the prediction system, there by increasing the speed and efficiency.

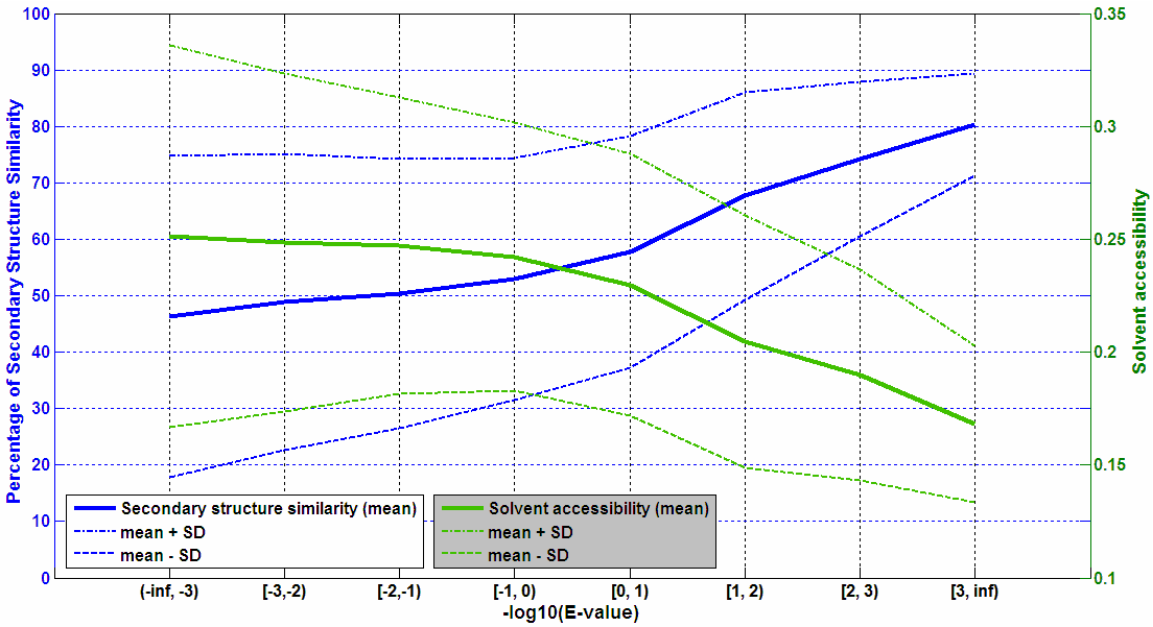


Figure 2.9: The plot showing the relationship between the log transformed E-value, secondary structure similarity and the mean absolute error of the solvent accessibility. The secondary structure similarity on the vertical axis on the left is plotted with negative logarithm of E-value on the horizontal axis. The mean absolute error of the solvent accessibility is plotted using the vertical axis on the right and the log-transformed E-value on the horizontal axis. SD stands for standard deviation in each bin.

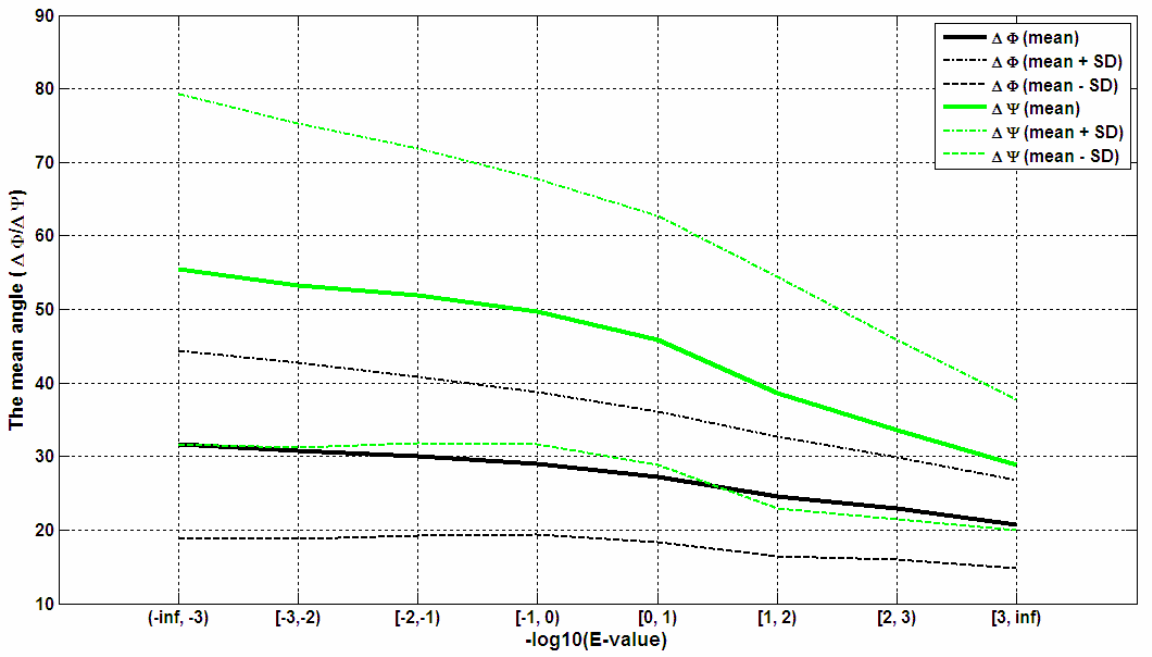


Figure 2.10: The mean deviation of Φ and Ψ angles in the alignment regions are plotted against the negative logarithm of E-value.

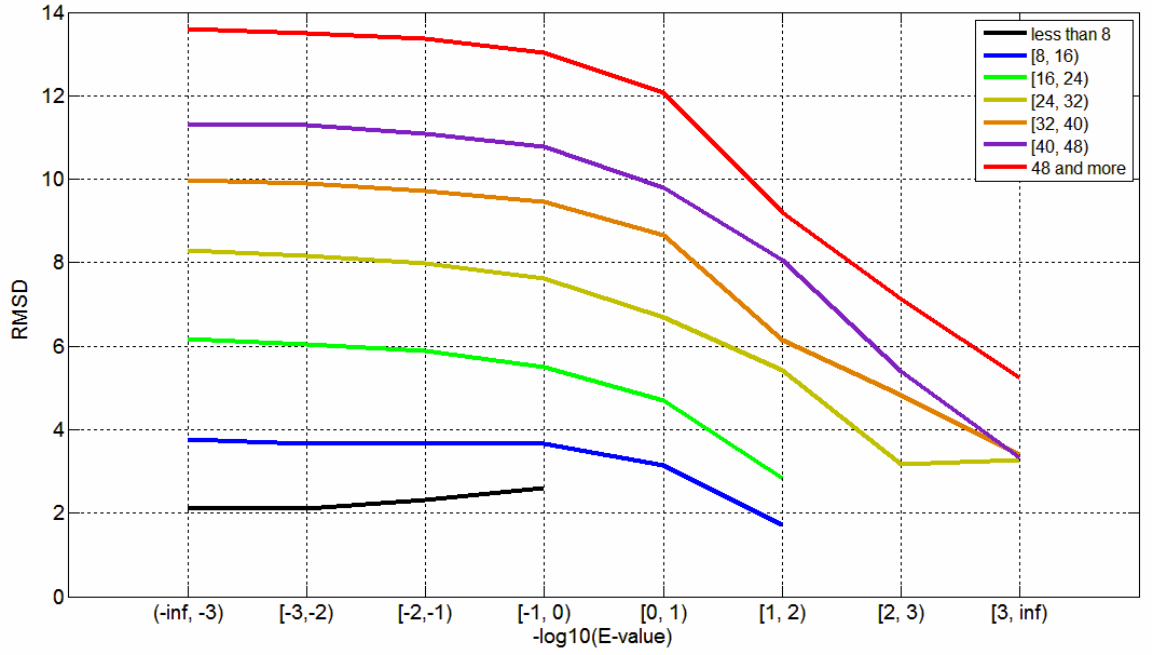


Figure 2.11: The plots depicting the relationship between the log-transformed E-value and the RMSD of the structural alignment for a BLAST hit. The hits of alignment length less than 8 were found to have no correlation between RMSD and the E-value.

We now discuss the effect of parameters of the PSI-BLAST algorithm. Specifically, we will discuss the effect of substitution matrices and the number of iterations of PSI-BLAST to run for building the profile. A substitution matrix contains the scores for amino acid substitutions. The score is proportional to the probability that amino acid *i* mutates into amino acid *j* for all pairs of amino acids. These matrices are constructed by performing statistical analysis of large number of sequences. The number of sequences is large enough to be statistically significant and the resulting matrices reflect the probabilities of mutations occurring through evolution. Two types of substitution matrices are used by the BLAST program, i.e., the PAM [Dayhoff *et al.*, 1978] and BLOSUM [Henikoff and Henikoff, 1992] matrices. PAM (Percent Accepted Mutation) substitution matrix is a look-up table in which scores for each amino acid substitution have been calculated based on the frequency of that substitution in closely related proteins that have experienced a certain amount of evolutionary divergence. BLOSUM stands for Block Substitution Matrix. It is a scoring matrix in which the substitution values are derived from the frequencies of substitutions in blocks of local alignments in related proteins. There are different BLOSUM and PAM matrices. We choose three matrices from each of PAM and BLOSUM matrices that represent the spectrum of these matrices. In PAM, we experimented with PAM250 (liberal of all PAM matrices, that allows for approximate alignments), PAM70 (facilitates alignments of intermediate homology) and PAM30 (facilitates exact alignments). In BLOSUM, we experimented with BLOSUM45 (allows liberal local alignments), BLOSUM62 (the most widely used substitution matrix, that produces intermediate local alignments) and finally

BLOSUM90 (favors exact local alignments). We present the effect of the substitution matrix on the secondary structure similarity. The plots are depicted in Figure 2.12. We next changed the number of iterations used to build the profile by PSI-BLAST. Once again, the effect of this parameter is studied on the secondary structure similarity. The plots are illustrated in Figure 2.13. It is not obvious from these plots which PSI-BLAST parameters are the best. We went further to analyze the number of hits in different bins of E-value using each of the parameter. As the previous plots showed, the lower the E-value, the more informative the alignments are. We therefore examined the number of hits in the more informative regions. The distribution of hits in various bins with the substitution matrix is presented in Table 2.1. Clearly, the hits generated with BLOSUM90 have more hits in the significant bins. The distribution of hits in various bins with different number of iterations to build the profile is presented in Table 2.2. Looking at both the quality (significance) and quantity (number) of the hits in various regions, the optimal number of iterations to build the profile is 3.

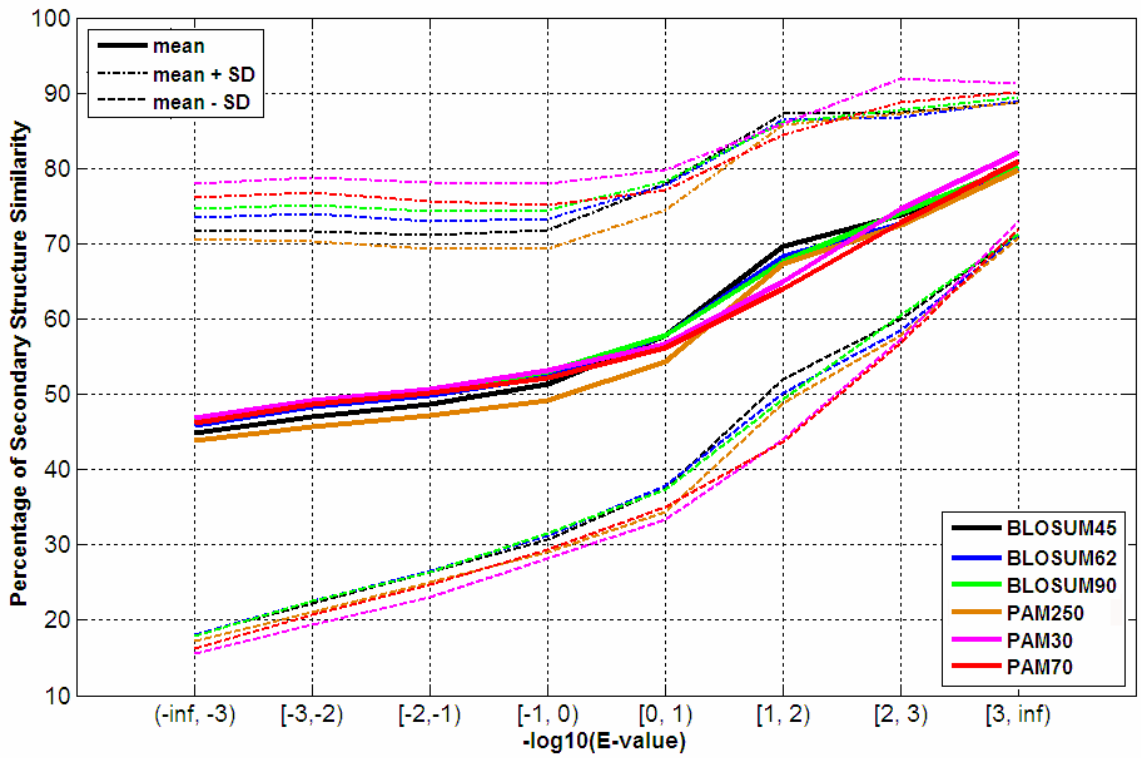


Figure 2.12: The effect of the various substitution matrices on the secondary structure similarity

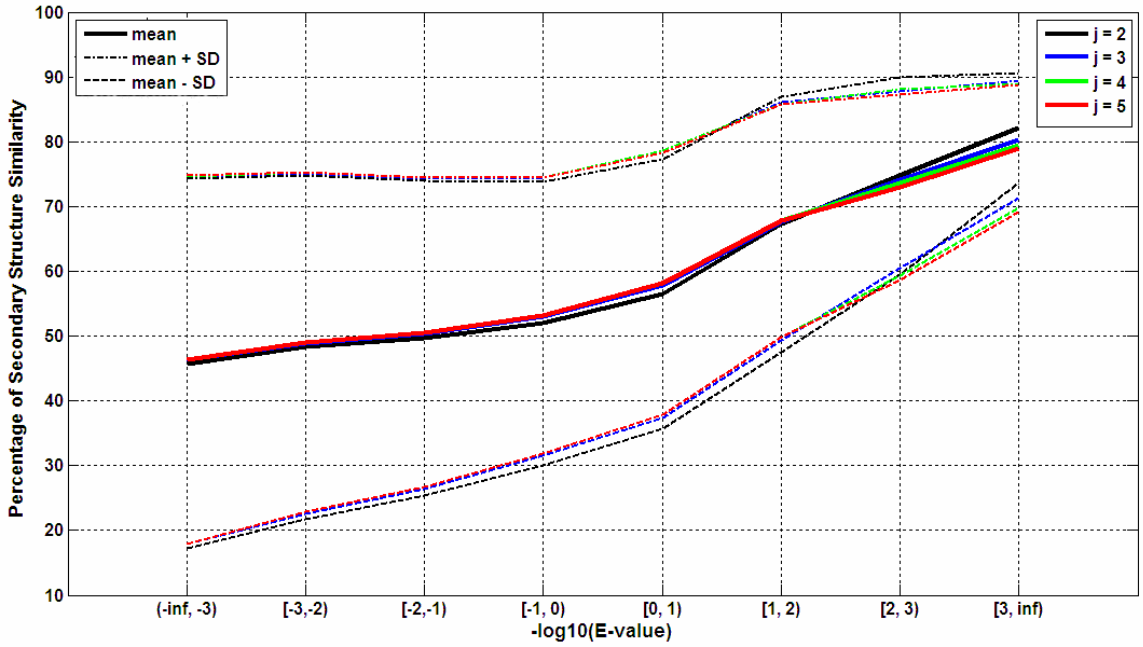


Figure 2.13: The effect of the number of iterations to build the profile on the secondary structure similarity

Table 2.1: The distribution of hits in various bins of significance for different substitution matrices

-log(E-value)	BLOSUM45	BLOSUM62	BLOSUM90	PAM250	PAM70	PAM30
(-∞, -3)	393,045	516,570	617,607	339,111	785,776	872,770
[-3, -2)	458,752	480,120	484,585	427,839	484,131	493,242
[-2, -1)	165,292	220,967	264,290	170,830	361,926	382,662
[-1, 0)	22,225	30,157	36,237	24,757	53,474	55,071
[0, 1)	3,142	4,151	4,876	3,563	7,373	6,953
[1, 2)	774	922	980	795	1,282	1,140
[2, 3)	329	389	432	320	381	307
[3, ∞)	2,150	2,384	2,345	1,925	2,094	1,453
Total	1,045,709	1,255,660	1,411,352	969,140	1,696,437	1,813,598

Table 2.2: The distribution of hits in various bins of significance for different number of iterations to build the profile

-log(E-value)	j = 2	j = 3	j = 4	j = 5
(-∞, -3)	688,746	617,607	598,175	591,848
[-3, -2)	490,830	484,585	484,189	483,400
[-2, -1)	282,166	264,290	256,946	254,565
[-1, 0)	37,050	36,237	35,627	35,331
[0, 1)	4,733	4,876	4,825	4,949
[1, 2)	883	980	1,070	1,096
[2, 3)	307	432	425	447
[3, ∞)	1,619	2,345	2,699	2,815
Total	1,506,334	1,411,352	1,383,956	1,374,451

2.3 Our Framework

The qualitative and quantitative analysis presented in the previous section indicates that the PSI-BLAST program is a good choice for searching for homologous fragments for a given query sequence. We were also able to experimentally discover optimal parameters

for running the PSI-BLAST programs. With these results in hand, we implemented a suite of programs to retrieve the compatible fragments and the associated structural information for a given protein sequence.

The profile of the query protein is first calculated using *nr* database with the PSI-BLAST program. The following parameters are employed to generate the profile: *j*=3, *e*= 0.001, and M=BLOSUM90. The other parameters are left to their default values. The profile is then used for searching the representative protein sequence database for compatible fragments. During profile-sequence alignment, the threshold value of *e* was set to 11,000 when searching the RPS. The main components of the framework are three programs and three databases. These programs, in order of their usage are PSI-BLAST, 'BLAST output parser' and the 'DSSP parser'. The output (default output format) of the PSI-BLAST program is processed by the 'BLAST output parser'. For all the hit fragments, the program collects the alignment statistics and structural information by calling 'DSSP parser'. The 'DSSP parser' program collects the information from the DSSP file of the hits, at the request of 'BLAST output parser'. It also takes care of the mapping of eight secondary structure classes to three secondary structure classes and also the calculating the relative solvent accessibilities from their absolute solvent accessibilities. The PSI-BLAST program is the most important and most time consuming of all the three programs. On average, PSI-BLAST takes about 0.55 second/residue [Bondugula and Xu, 2007] and the remaining part takes as little as 3 seconds. All of these programs are implemented in C/C++. Given a query sequence, the framework outputs the following information into plain text files:

1. profile
2. homologous sequence fragments
3. alignment length
4. alignment score
5. amino acid identity percentage in the alignment region
6. amino acid similarity percentage in the alignment region
7. raw scores
8. E-values
9. secondary structures of the homologous fragments
10. relative solvent accessibilities
11. dihedral angles (Φ and Ψ)
12. Cartesian co-ordinates

The block diagram of our framework is depicted in Figure 2.14.

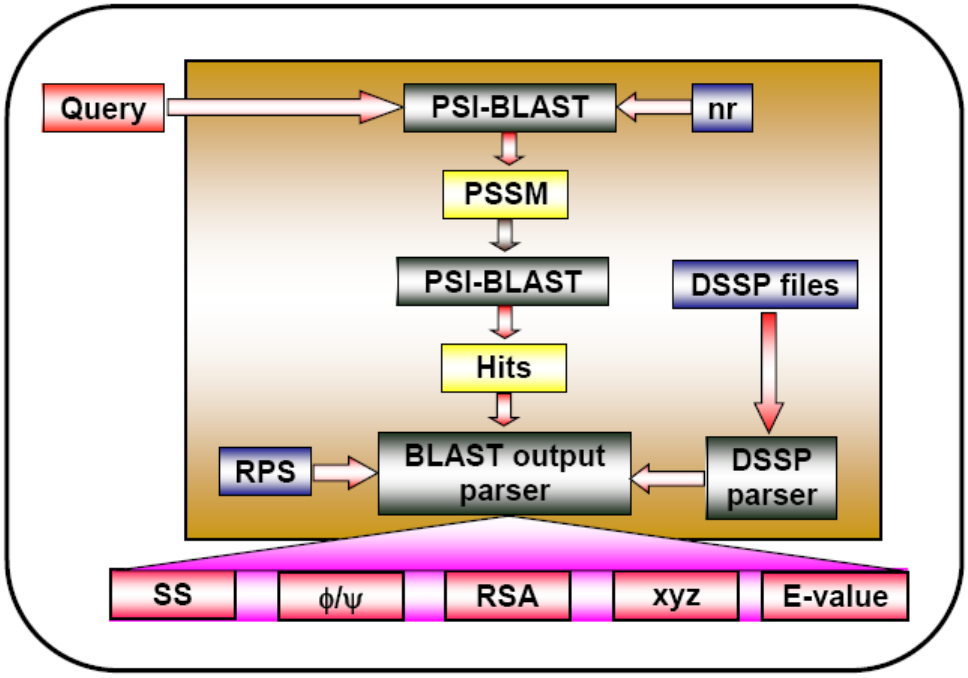


Figure 2.14: Our framework for protein structure prediction. The profile of the query protein is first calculated using *nr* database with the PSI-BLAST program. The profile is then used for searching the representative protein sequence database for compatible fragments. These fragments are parsed and the structural information of these fragments is collected from their respective DSSP files. The framework contains three programs (grey) and three databases (blue).

3. Application of Our Framework for Protein Secondary Structure Prediction

3.1 Introduction

Protein secondary structure is defined by the conformation of protein backbone. The backbone of a protein or peptide consists of repeated units with the amide nitrogen N(H), the carbon Cα, and the carbonyl carbon C(═O). An α-helix is a major secondary structure, which is almost always right handed as found in the threads of standard wood screw. A helix is formed when the hydrogen in the N─H of the n^{th} amino acid makes a hydrogen bond with oxygen in the C═O of the $(n+4)^{th}$ amino acid. This pattern of repeated bonding results in a stable α-helix. On average, there are 3.6 amino acids per turn in an α-helix. Other varieties of helices exist with slightly more or slightly less amino acids per turn. The schematics of an α-helix is illustrated in Figure 3.1.

The second major type of secondary structure is β-strand (see Figure 3.1). In a β-strand, usually 5-10 consecutive amino acids are in almost fully extended conformation. When more than one β-strand lie adjacent in space, a pleated β-sheet is formed. These are held by the hydrogen bonding between C═O groups of one strand and the N─H of the adjacent strand. If all the strands in a β-sheet run in the same biochemical direction from the start (amino terminal) to the end (carboxy terminal) of the protein, parallel β-sheets are formed. If alternating strands in the β-sheet run in opposite directions, anti-parallel sheets are formed.

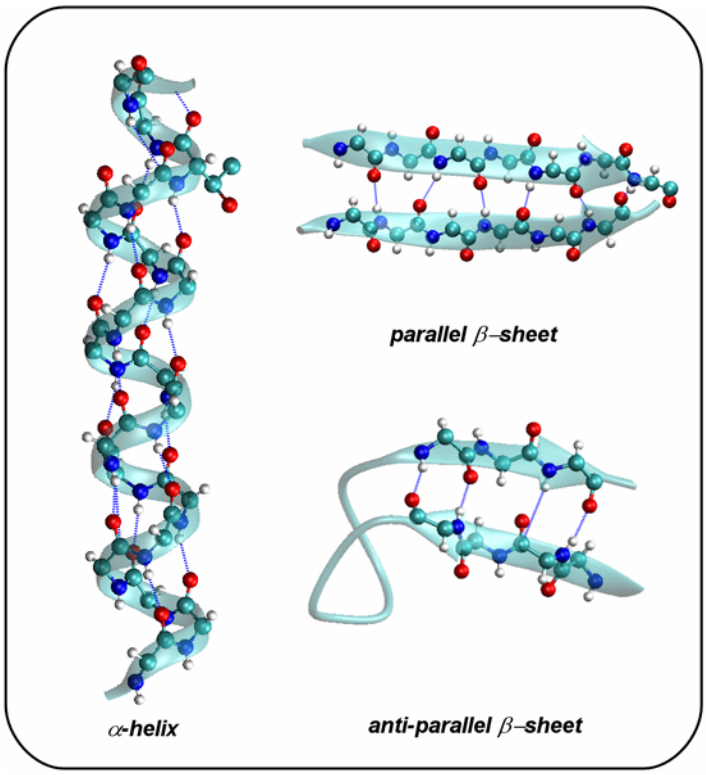


Figure 3.1: Protein secondary structure. A α-helix is formed when hydrogen bonds (blue) are formed between the hydrogen of N-H in residue 'n' and the oxygen of C=O of the residue 'n+4'. The β-sheets are held by the hydrogen bonds between the hydrogen atoms of N-H of one strand and the oxygen atoms of C=O of an adjacent strand in space.

In a folded protein, each amino acid adopts one of the following eight secondary structure classes: H (α-helix), G (3_{10}-helix), I (π-helix), B (isolated β-bridge), E (β-strand), S (bend), T (turn), and C (rest). Generally, researchers focus on a simplified version of the problem that contains only three secondary structure classes (see Chapter 2). Given an amino acid sequence, the aim of protein secondary structure prediction is to computationally assign each residue into one of the three secondary structure classes. An example is illustrated in Figure 3.1.

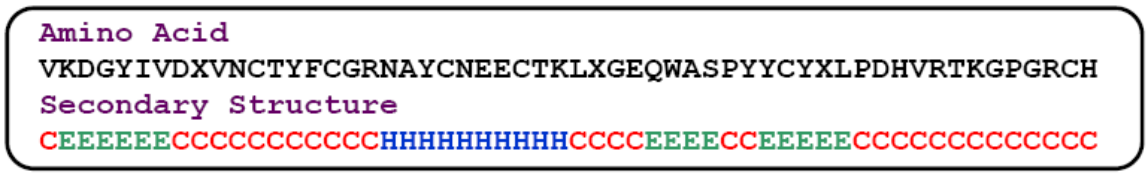


Figure 3.2: A protein fragment and its corresponding secondary structure.

Secondary structure prediction plays an important role in characterizing protein structures and providing a basis for tertiary structure prediction [Rost, 2001; Meiler and Baker, 2003]. The secondary structure of the protein provides the computational methods with constraints that reduce the search space greatly and therefore making the prediction more efficient and faster. Predicting the secondary structure of the protein before predicting the tertiary one mimics the natural order of events in the folding pathway, i.e., the secondary structure formation is followed by folding the protein into a three-dimensional compact structure. Therefore, the study of secondary structure prediction is a crucial part in protein three-dimensional structure prediction.

3.2 Secondary Structure Prediction

MUPRED [Bondugula and Xu, 2007] is the secondary predicted system that is based on our framework. It mainly consists of a neural network that uses two types of features. The first type of features is class membership values at each residue position generated by fuzzy k-nearest neighbor (FKNN) algorithm, a generalized form of KNN method, while the second type of features is normalized PSSM. The output of our framework provides the necessary information for MUPRED for feature generation. In the following sub-

sections, we explain the secondary structure specific components that are part of MUPRED in addition to our framework.

3.2.1 Fundamental Algorithm

Currently, the most successful methods depend on machine learning techniques such as neural networks [Holly and Karplus, 1989; Rost and Sander, 1994; Chandonia and Karplus 2005; Jones, 1999; Baldi et al., 1999], nearest neighbor methods [Salamov and Solovyev, 1995; Salamov and Solovyev, 1997; Bondugula *et al.*, 2005] and hidden Markov models [Karplus *et al.*, 1998].

In neural network methods, each amino acid of the query sequence is represented either by a 21-dimension binary vector such that only the dimension corresponding to the current amino acid is one and the rest are zero. Recent methods incorporate multiple sequence alignment information by using the column corresponding to the current amino acid in the PSI-BLAST profile. A sliding window scheme that includes neighbors on the both sides is used to classify each amino acid. Basic methods use a single feed forward networks, while more advance methods use multi-stage neural networks. In the nearest neighbor methods, the PSSM of the query protein is first divided into rectangular blocks, using a sliding window scheme. Each block represents an amino acid in the query protein. The dimension of each block is 21xW, where W is dependent on the number of neighbors to consider on each side (if n neighbors are considered, W = 2n+1) and the 21 rows have the same interpretation as in PSSM, i.e., each position represents the propensity of amino acids to be found at that particular position. The position weight is such that the center position receives the highest weight and the weight gradually

decreases on each side of the center position. In order to predict the class of each amino acid, k blocks that represent the amino acids with known solvent accessibilities are selected form the database. These k blocks are selected such that the sum of the position-weighted, element wise differences of the elements in the block (distance measure) is lowest of all other blocks. Using the classes of neighbors and their distances from the block representing the current residue are used to predict the state of the current residue using the nearest neighbor algorithms. HMM methods create a hidden Markov model from a single target sequence by iteratively finding homologs in a protein database and refining the model. A hidden Markov model is a sequence of nodes that correspond to a column in a multiple sequence alignment. The three states used by the model are match state, insert state and a delete state. Each position has a distribution of bases, as do transitions between states. That is, these linear HMMs have position-dependent character distributions and position-dependent insertion and deletion gap penalties. The alignment of each of a family to a trained model automatically yields a multiple alignment among those sequences. Each sequence can be represented by a series of such states. In many ways, these models correspond to profiles. For a given query sequence, the HMM methods attempt to find and multiply align a set of homologs and then create an HMM from that multiple alignment. The resulting HMM is then used for database search and the state of the each residue is predicted based on the matches in the databases.

Each of these techniques has its own strengths and weaknesses. For example, in general, neural networks perform better in case where the complex patterns recognition is involved. Similarly, hidden Markov models are good at capturing the first order

relationships between the various states of the models. K-nearest neighbor (KNN) methods are sub-optimal methods and the 1-NN rule is bounded above by no more than twice the optimal Bayes error rate [Keller *et al.*, 1985]. Nearest neighbor models are simple and transparent models that do not require retraining whenever new data is available. On the other hand, the systems based on neural network methods and the hidden Markov models perform well if the test data is similar to the training data, specifically in the context of protein secondary structure prediction, the systems perform well if query protein has homologs in the database [Zhang *et al.,* 1992; Yi & Lander, 1993], where as some models based on nearest neighbor methods are not limited by absence of homologs in the database.

Hybrid models provide us with methods to combine the strengths of the individual methods and overcome their weaknesses to some extent. In this chapter we introduce a hybrid method for protein secondary structure prediction, in which we closely integrate the FKNN and the neural network method into the same system to provide a balanced prediction for both the queries with homologs in the database and the queries without homologs in the database. The server can be accessed by general public at *http://digbio.missouri.edu/mupred*.

3.2.2 Scoring Scheme

The compatible fragments returned by our framework are scored using the following equation:

$$S = \max\{1, 7 + \log_{10}(\text{Evalue})\} \tag{3.1}$$

The above expression was designed so that it roughly emulates the notion of the 'dissimilarity'. Matching fragments whose similarities to the segments of query sequence are statistically significant have high expectation values and therefore low scores. Similarly, for matching fragments whose similarities are not significant, the scores are high.

3.2.3 Databases

The RPS described in Chapter 2 was divided into two parts. The first part is used for training MUPRED and the second, for testing. The proteins were sorted according to the PDB release dates. We chose the oldest (according to the PDB release dates) 1,000 proteins for tuning the FKNN algorithm and for training the neural networks. The latest 200 proteins in this database were used as the first benchmark dataset to test and compare the performance of MUPRED with other secondary structure prediction systems. The training proteins contained 335,531 residues with 35.14% Helix residues, 23.75% Strand residues and 39.43% Coil residues. We used the Astral SCOP [Brenner *et al.*, 2000] protein domain database version 1.69 to derive a second protein set for benchmarking purposes. Each protein sequence of the original database, which contained 5,457 protein domains, was searched for homologs in the training sets of MUPRED and other prediction software. If a homolog was found with a statistical significance value (E-value) of less than or equal to 0.1, the query sequence was discarded from the benchmark set. Similar to earlier dataset, protein domain sequences that are shorter than 40 residues were removed and sequences that are composed of less than 90% of regular amino acids are discarded too. After this filtration process, only 1,934 domain sequences remained in

the second benchmarking protein set. The authors preferred the above method to evaluate and compare the performance of MUPRED with existing software to standard cross-validation schemes for the following reason: the earlier methods did not have access to large numbers of proteins, both for building the PSSM and the training data sets.

3.2.4 Method

MUPRED incorporates PSSM of the query protein for secondary structure prediction through PSI-BLAST and the *nr* database. The PSSM returned by our framework is used to generate the first set of features. We converted the PSSMs into vectors that are suitable for training the neural networks. First, these values were scaled into [0 1] using the maximum and minimum in the PSSMs of all the proteins in the database. Each position in the query sequence is represented by a 20-dimensional vector representing the likelihood of each amino acid occurring at that position. An additional bit is used to mark the termini of the protein, resulting in a 21-dimensional vector per position. These scaled PSSM values are converted into vectors suitable for neural networks using the sliding window scheme, i.e., the vector that represents the profile values of the current residue is flanked by its neighbors on the both sides. The rationale for this process is that the secondary structure of an amino acid is not only based on the current amino acid, but also on its neighbors. The number of residues that will be flanked on each side is determined by the window size *W*. The authors experimentally found that *W*= 13 worked the best. Therefore, the first feature set consists of 21x13 = 273 features per residue.

The second set of features is generated from the compatible fragments returned by our framework. These fragments are treated as nearest neighbors. The secondary

structures associated with the fragments are used as the labels of these neighbors. These labeled neighbors are then used to calculate the membership value of the current residue in three classes. These membership values represent the confidence with which the current residue belongs to the three secondary structure classes. Figure 3.3(a) illustrates the database fragments for a typical query protein. The highlighted column depicts the neighbors using the multiple sequence alignments of the hits with the query protein. The secondary structures, E-values and the scores corresponding to the database fragments are displayed in Figure 3.3(b), 3.3(c) and 3.3(d) respectively.

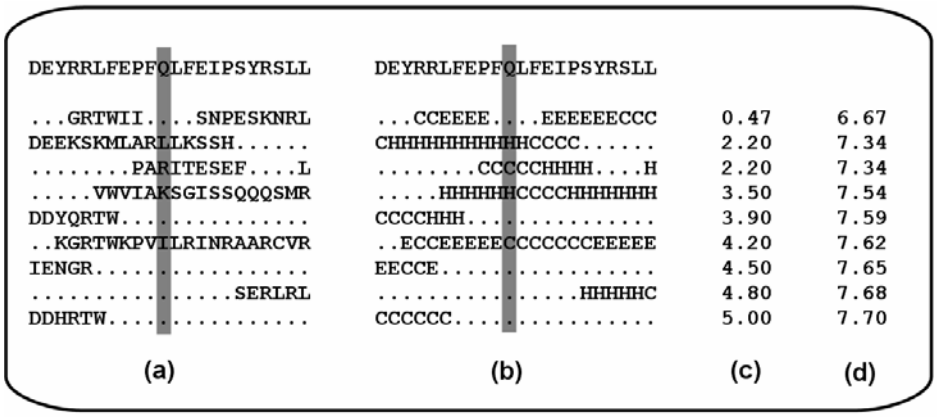


Figure 3.3: Calculation of the membership value of each residue in secondary structure classes. The query protein is shown in the top row. (a) Database fragments from the PSI-BLAST matches; (b) corresponding secondary structures of the database matches; (c) corresponding expectation value of the hits; (d) scores of the hits calculated from their respective expectation values.

3.2.5 Nearest Neighbor Method

Given a set of feature vectors $X = \{x_1, x_2,\ldots, x_n\}$, $x_j \in R^d$ and their corresponding class labels u_{ij}, i = 1, 2, …, C the task of the nearest neighbor algorithm is to find the class

label of a new vector y, using the labeled data X. In the crisp case, the u_{ij} has '1' in only one class and '0' every where else, i.e., the constraints $u_{ij} \in \{0,1\}$ and $\sum_{i=1}^{C} u_{ij} = 1$ for each j, hold. By relaxing the first constraint that a feature can have real membership value in [0 1], we obtain the fuzzy nearest neighbor algorithm. Relaxing the second constraint is also possible and it results in possibilistic nearest neighbor algorithm. In the crisp nearest neighbor algorithm, given a vector x, the nearest k neighbors are found based on some predefined distance metric. The x is assigned to a class to which majority of neighbors belong to. In the current work, we use the FKNN algorithm to predict the secondary structure of each residue. The rationale behind the choice of FKNN is explained later in the paragraph.

The secondary structure state of each residue can be predicted from class membership values of the neighbors with the FKNN algorithm. The following technique adopted and modified from [Keller et al., 1985] provides the procedure to calculate the membership values of the current residue from the labeled neighbors. Let $P = \{r_1, r_2, ..., r_l\}$ represent a protein with l residues. Each residue r has k-nearest neighbors, i.e., hit fragments that have a residue aligned with the current residue (see Figure 3.3). Also, let u_{ij} be the membership in the ith class ($i \in \{Helix, strand, Coil\}$) of the j^{th} neighbor. For each r, the predicted membership value u_j in class i can be calculated using the following algorithm:

BEGIN

Initialize $i = 1$.

DO UNTIL (r assigned membership in all classes)

Compute $u_i(r)$ using:

$$u_i(r) = \frac{\sum_{j=1}^{k} u_{ij}\left(1 / \left\| S(r_j)^{2/m-1} \right\|\right)}{\sum_{j=1}^{k}\left(1 / \left\| S(r_j)^{2/m-1} \right\|\right)}, \tag{3.2}$$

Increment i.

END DO UNTIL

END

It can be noticed from Equation 3.2 that the contribution of each neighbor (hit r_j) in the calculation of membership value of the current residue in each class is determined by the score *S*, which in turn is determined by the significance of the hit retuned by our framework. The influence of the score can be controlled by the fuzzifier '*m*' [Keller et al., 1985]. If the value of fuzzifier is set to 1.5, the class membership value of the residue is proportional to the inverse of the fourth power of score and so on. In this case, we experimentally found that $m = 1.5$ yields the best results. For each position, there are three numbers indicating how much each residue belongs to each of the three secondary structure classes according to the FKNN algorithm. Similar to the first feature set, a sliding window with W = 11 was used to generate the second feature set. They also included an additional bit to mark the end of the protein. This feature set therefore consists of vectors that contain11×4 = 44 features per residue.

A neural network is used to integrate the information from the normalized profiles and the FKNN algorithm. The network is a fully connected feed forward network with one hidden layer. The features are fed into the input layer. The hidden layer consists of 300 units and was experimentally determined. The output layer consists of three nodes, one for each of the Helix, Strand and the Coil classes. The final architecture of the network is as follows: (273+44)×300×3 (input nodes × hidden nodes × output nodes). The values generated by the output nodes are the final class membership values of the current residue in each of the three secondary structure classes. We trained 100 networks and use the average value of the top four networks to determine the membership values. The block diagram of the MUPRED is depicted in Figure 3.4.

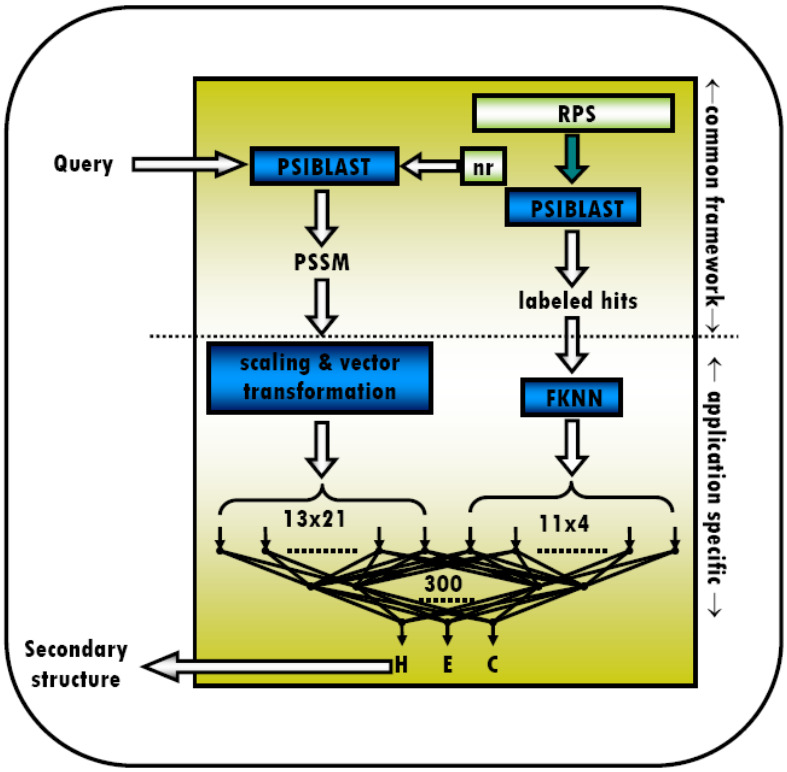


Figure 3.4: The block diagram of the MUPRED protein secondary structure prediction system. The profile of the query protein is used to generate two types of features. The first feature set consists of fuzzy class memberships of each residue in the three secondary structure classes. The second feature set consists of normalized profile. The features are transformed into vectors suitable for neural network training using a sliding-window scheme of window length W. For the profile-derived feature-set, W=13 is used. An extra bit is used to mark the termini of each protein. The PSSM feature-set, therefore, consists of 13x21=273 features. For the fuzzy memberships, W=11 is used and, similar to the PSSM feature set, an extra bit is used to mark the termini of the protein, resulting in 11x4=44 features. The dotted separates the components of the system that are contributed by the common framework and the components that are specific to protein secondary structure prediction system.

The advantage of FKNN over the traditional (crisp) KNN algorithms is that residues are assigned a membership value in each class rather than binary decision of ‘belongs to’ or ‘does not belong to’. Such an assignment allows us to use these membership values as (quantitative) strength or confidence with which the current

residue belongs to a particular class. These strengths when fed to neural network along with the PSSM resulted in better performance when compared with existing methods. In the traditional (crisp) KNN, all the neighbors are weighted equally, which is not necessarily true in the context of proteins i.e., some protein fragments are more similar to the sub-sequences of the query protein than other fragments. This similarity is captured in the formulation of expectation value (E-value), which in turn is transformed in to score 'S' in MUPRED. FKNN was formulated such that these relative distances (score S in this case) are weighted while the query vector (current amino acid) is classified (into one of the three secondary structure classes). Another advantage of using FKNN over the traditional crisp version is that the membership values of the residues along the sequence of the protein show a smooth transition from state to another, accurately representing the state transitions in real proteins. The superiority of FKNN over the traditional KNN algorithm for protein secondary structure prediction was also demonstrated in earlier work that lead to the development of MUPRED [Bondugula et al., 2005].

3.3 Results

There are two popular methods to measure the accuracy of secondary structure prediction systems. They are *Q*-measures [Rost and Sander, 1994a] and Matthew's correlation co-efficient [Matthews, 1975]. We used these two measures to evaluate the performance of MUPRED and compared it with other existing software. The *Q*- measures are defined as follows:

$$Q_3 = \frac{C}{T}, \tag{3.3}$$

$$Q_{structure} = \frac{C_{structure}}{T_{structure}}, \tag{3.4}$$

where C is the number of amino acids correctly classified in all three classes, T is the total number of amino acids, structure is one of {Helix, Strand, Coil}, C_{Helix} is the number of amino acids in Helix configuration that are correctly classified, while T_{Helix} is the total number of amino acids in the Helix configuration and so on. The Matthew's correlation coefficients are defined as follows:

$$M_{structure} = \frac{TP.TN - FP.FN}{\sqrt{(TN + FN)(TN + FP)(TP + FN)(TP + FP)}}, \tag{3.5}$$

where structure is one of {Helix, Strand, Coil}, TP is the number of positive cases that are correctly predicted, TN is the number of negatives that are correctly rejected, FP is the number of false positive cases and FN is the number of false negative cases. For example, if residue in Helix is correctly predicted as Helix then it is a true positive case. If a non-Helix (either Strand or Coil) residue is correctly predicted as a non-Helix, then it is the case of true negative. If a Helix residue is predicted as a non-Helix residue, then it the case of false negative. Finally, if a non-Helix residue is predicted as Helix residue, it is case of false positive.

We compare the performance of MUPRED with PSIPREDv1 [Jones, 1999] and SSPro4 [Baldi *et al.*, 1999]. Both of them use PSSMs and neural networks and were also

trained on the training sets that contained similar number of sequences as in training set for MUPRED. The performance of the MUPRED version that contained only FKNN algorithm (PSSM is used only to search the database) followed by a neural network filter is also reported. We present these results in Table 3.1. The prediction accuracy of our method on the training set is presented in Table 3.2.

Table 3.1: The performance comparison of various algorithms on the two benchmark sets.

Algorithm	Test set	Q_3	Q_{Helix}	Q_{Strand}	Q_{Coil}	M_{Helix}	M_{Strand}	M_{Coil}
FKNN+NN	B1	73.9%	76.2%	67.2%	76.1%	0.66	0.61	0.54
MUPRED	B1	79.2%	80.9%	72.4%	82.0%	0.74	0.69	0.62
PSIPREDv1	B1	75.9%	78.4%	68.3%	78.6%	0.70	0.63	0.56
SSPro4	B1	77.4%	82.7%	66.7%	79.5%	0.73	0.65	0.59
FKNN+NN	B2	76.1%	80.0%	68.2%	76.8%	0.69	0.63	0.57
MUPRED	B2	80.1%	83.9%	72.6%	80.8%	0.75	0.69	0.63
PSIPREDv1	B2	77.1%	80.2%	68.3%	79.0%	0.72	0.63	0.58
SSPro4	B2	78.4%	84.4%	67.3%	79.0%	0.74	0.65	0.60

Q_3 is the fraction of amino acids whose secondary structures have been accurately predicted in all three classes. Q_{Helix}, Q_{Strand} and Q_{Coil} are the fraction of amino acids that are accurately predicted in Helix, Strand and Coil classes respectively. Similarly, M_{Helix}, M_{Strand} and M_{Coil} stand for Matthew's correlation coefficient for Helix, Strand and Coil classes respectively. B1- the 200 protein benchmark set derived from the March 2006 release of PDBSelect database. B2- the 1934 protein domain benchmark set derived from Astral SCOP database version 1.69.

Table 3.2: The performance of FKNN+NN system and the MUPRED prediction system on the 1798 training protein set.

Algorithm	Q_3	Q_{Helix}	Q_{Strand}	Q_{Coil}	M_{Helix}	M_{Strand}	M_{Coil}
FKNN+NN	74.93%	77.45%	68.06%	76.83%	0.68	0.62	0.55
MUPRED	81.19%	83.04%	74.22%	83.76%	0.77	0.72	0.65

In order to assess the quality of the predictions, we divided the final membership values generated by MUPRED for the proteins in the test protein set into bins in each class, as shown in the third column of Table 3.3, such that the average probability that the given prediction is accurate falls into intervals shown in the fourth column. Depending on the final membership values in each class, a confidence score (shown in column 5 of Table 3.3) will be assigned to that prediction. This confidence values enable the users to identify the regions of the protein for which the prediction is more likely to be accurate. For a given class membership value, the probability that the prediction is accurate can be looked up using the plot in Figure 3.5 or the values in Table 3.3. For example, if the confidence value is 4 for a predicted helix, we know that the probability for this prediction to be true is 75%-80%.

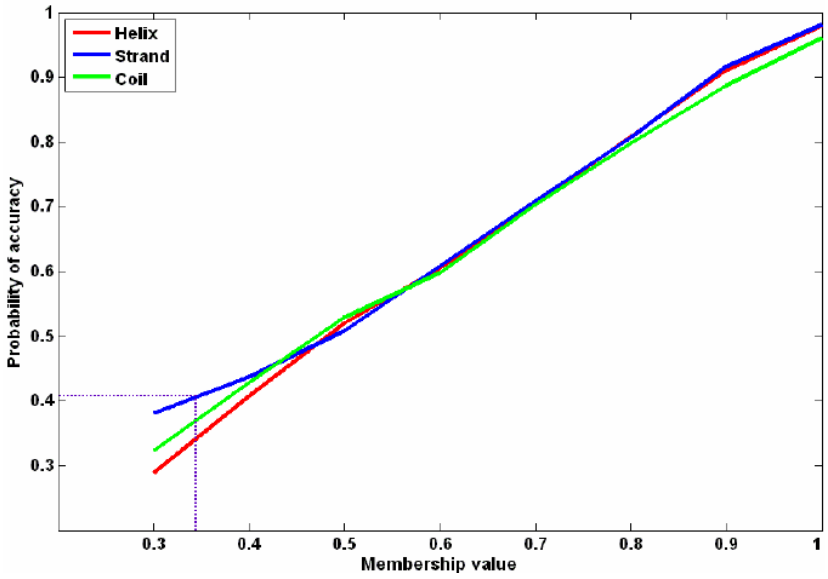


Figure 3.5: The probability for the prediction of a secondary structure to be accurate vs. class membership value.

Table 3.3: Confidence value to assess the quality of a prediction. Column 1: secondary structure class; column 2: number of samples that were used to derive the confidence values; column 3: the range of the class membership values to which the confidence corresponds to; column 4: the range of probabilities within which the predicted structure is accurate and column 5: the output confidence value that will be assigned to the residue.

secondary structure class	number of samples	membership values	probability of accurate prediction	assigned confidence
Helix	301	[0.00 0.417)	[0.0 0.3)	0
	1689	[0.417 0.629)	[0.3 0.55)	1
	823	[0.629 0.707)	[0.55 0.65)	2
	1300	[0.707 0.812)	[0.65 0.75)	3
	762	[0.812 0.872)	[0.75 0.85)	4
	1060	[0.872 0.915)	[0.85 0.90)	5
	2060	[0.915 0.965)	[0.90 0.95)	6
	201	[0.965 0.969)	[0.95 0.98)	7
	2309	[0.969 0.991)	[0.98 0.99)	8
	1176	[0.991 1.000)	[0.99 0.996)	9
Strand	322	[0.00 0.422)	[0.0 0.3)	0
	1206	[0.422 0.590)	[0.3 0.55)	1
	1257	[0.590 0.711)	[0.55 0.65)	2
	58	[0.711 0.717)	[0.65 0.75)	3
	1265	[0.717 0.832)	[0.75 0.80)	4
	2200	[0.832 0.954)	[0.80 0.90)	5
	100	[0.954 0.958)	[0.90 0.95)	6
	700	[0.958 0.976)	[0.95 0.96)	7
	633	[0.976 0.988)	[0.96 0.97)	8
	1218	[0.988 1.000)	[0.97 0.985)	9
Coil	240	[0.00 0.355)	[0.0 0.43)	0
	1688	[0.355 0.501)	[0.43 0.50)	1
	4587	[0.501 0.687)	[0.50 0.60)	2
	877	[0.687 0.717)	[0.60 0.70)	3
	5455	[0.717 0.880)	[0.70 0.80)	4
	2580	[0.880 0.951)	[0.80 0.90)	5
	133	[0.951 0.955)	[0.90 0.94)	6
	200	[0.955 0.963)	[0.94 0.975)	7
	467	[0.963 0.992)	[0.975 0.98)	8
	400	[0.992 1.000)	[0.98 1.00)	9

The program was written in the ANSI standard compatible C programming language. The Windows version of the program was tested on a desktop PC with 3.2 GHz Intel Pentium-4 processor and 2 GB RAM. On average, the time required to predict the secondary structure was found to be 0.55 sec/residue, including running PSI-BLAST.

3.4 Summary

MUPRED, based on our novel framework bridges the gap between the template based methods that find alignments between the whole query sequence or its short fragments and sequences in the protein structure database PDB and sequence profile based methods in which the sequence profile is derived from the similar sequences (typically without structural information). Template based methods are successful when sequences similar to the query sequence can be found in PDB, but have limited performance otherwise, mainly due to lack of using sequence profile information of the query protein. In contrast, sequence profile based methods take advantage of the sequence profile information but use the structure information in PDB indirectly. MUPRED, using the framework, overcomes this limitation by looking for fragments in the database that are similar to the segments of the query sequence rather than sequence-level homologs. Integrating these two fundamentally different models into a single model enables MUPRED to provide balanced predictions for queries with or without homologs in the sequence database. The notable feature of MUPRED prediction system is that the accuracy of the prediction increases as more and more protein structures become available without retraining or retuning. The system also outputs the confidence values for each residue in the sequence,

which enables users to determine the regions where the prediction is more likely to be correct.

4. Application of the Framework for Solvent Accessibility Prediction

4.1 Introduction

While secondary structure captures some aspects of the protein structure, the solvent accessibility (SA) captures different kinds of features. The concept of the SA was introduced by Lee and Richards [1971] and may be defined as the extent to which the molecules of the solvent can access a residue or an atom of the proteins (Figure 4.1). The knowledge of SA of proteins helped to further the understanding of protein structure [Chotia, 1976; Janin 1976; Janin 1979; Wodak and Janin, 1980, Miller *et al.*, 1987; Eyal *et al.*, 2004, Yahyanejad *et al.*, 2006; Jacob and Unger, 2007], antigenic determinants [Thronton *et al.*, 1986; Novotny *et al.*, 1986;Huang *et al.*, 1990; Pavlink *et al.*, 2003; Kulkarni-Kale *et al.*, 2005], protein stability analysis [Saraboji *et al.*, 2005; Huang *et al.*, 2007;David *et al.*, 2007], protein structure classification [Gromiha and Suwa, 2003;Sujatha and Balaji, 2004, Yu *et al.*, 2006], protein interaction analysis [Ahmed *et al.*, 2004; Chen and Zhou, 2005; Hoskins et al, 2006], etc. For a given sequence, the target could be to predict either relative SA of a residue or as two or three state classification, in real value of area (Figure 4.1).

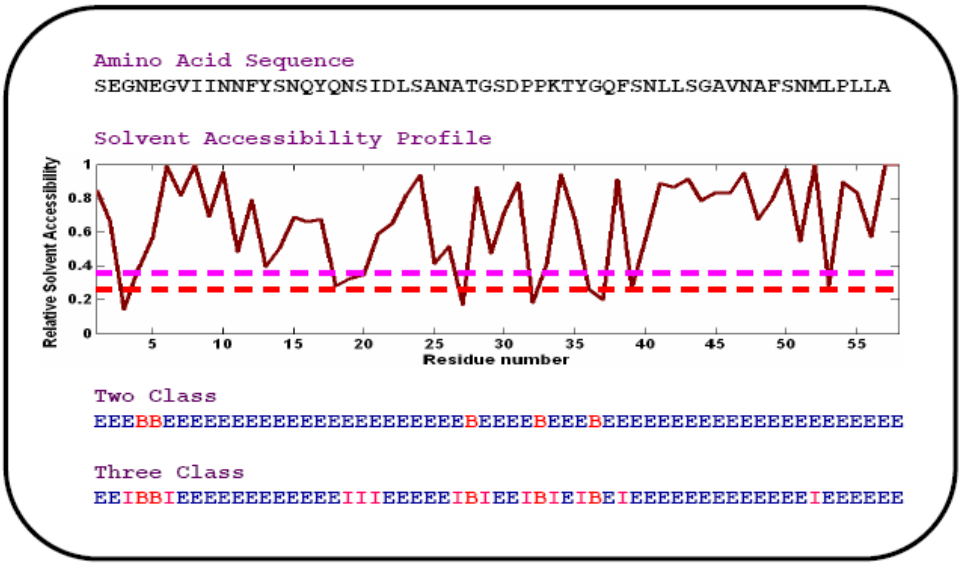


Figure 4.1: The protein solvent accessibility. Top- typical amino acid sequence. Middle- observed relative solvent accessibility profile. The pink and red dotted lines represent two possible thresholds for classification. Bottom- each residue is classified in to either a buried (B) or an exposed (E) residue. Sometimes, an additional class called intermediate (I) is also used. In the two-class classification illustrated above, the red dotted line is used as threshold to divide the buried and exposed classes. In case of three class classification, the red dotted line is used as a threshold for classifying buried and intermediate classes, while the pink dotted line is used to classify the intermediate and exposed classes.

4.2 Solvent Accessibility Prediction

Many approaches like support vector machines (SVM) [Yuan and Huang, 2004; Kim and Park, 2004], neural networks [Holbrook *et al.*, 1990; Rost and Sander, 1994; Ahmed and Gromiha, 2002; Adamczak *et al.*, 2004; Garg *et al.*, 2005], information theory [Manesh *et al.*, 2001] and nearest-neighbor methods [Sim *et al.*, 2005] have been proposed for SA prediction. Almost all of these methods rely on PSSM from multiple sequence alignments for the prediction of SA. There are at least two drawbacks of all these approaches. First, they all predict the structural features of the proteins without using the structural

information available in the PDB. Second, in case of proteins sequences that do not have close homologs in the database of known sequences (for example *nr*), the PSSM will not be well defined, making the predictions unreliable [Adamczak *et al.*, 2004].

SVM methods originate in computational learning theory and are based on structural risk minimization principle. In SVM, the samples are mapped in to high-dimensional feature space and an optimal hyper-plane that separates two classes is constructed. A small sample of the training points, known as support vectors, gives the hyper plane output and these vectors are closest to the hyper plane. Support vectors correspond to points that are hardest to classify. Each amino acid of the query sequence is represented either by a 21-dimension binary vector such that only the dimension corresponding to the current amino acid is one and the rest are zero. Recent methods incorporate multiple sequence alignment information by using the column corresponding to the current amino acid in the PSI-BLAST profile. A sliding window scheme that includes neighbors on the both sides is used to classify each amino acid. When protein solvent accessibility is classified into m states by m-1 cut-off thresholds, m SVMs are needed to predict m states. For each state, a SVM is trained on the samples of this state as positive and all samples of other states as negative. Neural networks methods use similar methods to generate the training vectors. The only difference is the underlying algorithm. Basic methods use simple feed forward networks, while more advance methods use recurrent neural networks. Some methods train multiple networks and use the average output of these networks as the final output. Some recent variants use multi-stage SVMs or neural networks. In information theory approach, the propensity of single-residue and

pair-wise residue interactions to adopt a conformational state are used. In order to use pair-wise residue interactions, the confirmation states of the neighboring residues are also used. In the nearest neighbor methods, the PSSM of the query protein is first divided in to rectangular blocks, using a sliding window scheme. Each block represents an amino acid in the query protein. The dimension of each block is 21xW, where W is dependent on the number of neighbors to consider on each side (if n neighbors are considered, W = 2n+1) and the 21 rows have the same interpretation as in PSSM i.e., each position represents the propensity of amino acids to be found at that particular position. The position weight is such that the center position receives the highest weight and the weight gradually decreases on each side of the center position. In order to predict the class of each amino acid, k blocks that represent the amino acids with known solvent accessibilities are selected form the database. These k blocks are selected such that the sum of the position-weighted, element wise differences of the elements in the block (distance measure) is lowest of all other blocks. Using the classes of neighbors and their distances from the block representing the current residue are used to predict the state of the current residue using the nearest neighbor algorithms.

Based on our framework, we propose a new method in which both the structural information and the sequence profile information are used. Unlike many approaches that classify each residue in either two or three classes based on predetermined thresholds, we predict the real solvent accessibility. The user may choose any threshold based on his/her specific needs, if the residues have to be classified into multiple classes. Also, most of the current methods were tested on small data sets containing up to a few hundred sequences.

These results on small sets vary a lot, depending on the kind of data set used. In order to overcome this problem, we tested our method on a large scale independent data set to measure the stable performance. The prediction program is implemented into the MUPRED package as a public web server at *http://digbio.missouri.edu/mupred* along with the secondary structure prediction server.

4.2.1 Fundamental Algorithm

We first build a structural profile by estimating the relative solvent accessibility of the query protein using fuzzy mean operator (FMO) from the solvent accessibilities of proteins with known structures. We then integrate the estimated SA and the PSSM using a neural network. The output of the neural network is the predicted relative solvent accessibility of the current residue.

The compatible fragments generated by our framework are scored based on the scoring function used in the secondary structure prediction application (Chapter 2). The relative solvent accessibility (RSA) of each residue of the query protein is calculated from the RSAs of hits that have a residue aligned with the current residue. The SAs of the hit fragments are calculated using DSSP program. For each residue, the absolute SA retuned by the DSSP program is transformed into RSA by dividing it with the maximum SA given in [Rost and Sander 1994]. The RSA of the query protein is calculated using the following expression for FMO:

$$RSA(r) = \frac{\sum_{j=1}^{K} RSA_j \left(\frac{1}{S^{\frac{2}{m-1}}} \right)}{\sum_{j=1}^{K} \left(\frac{1}{S^{\frac{2}{m-1}}} \right)}, \tag{4.1}$$

where r is the current residue, K is number of hits that have residue aligned with the current residue, RSA_j is the relative solvent accessibility of the residue in the j^{th} hit that is aligned with the current residue, S is score of the hit fragment, and m is a fuzzifier that controls the weight of the dissimilarity measure S. The optimal value of fuzzifier was experimentally determined to be 1.5. Note that the Equation (4.1) is a special case of the FKNN algorithm described in Chapter 3, it is classifier with only one class the class membership value is used as the predicted RSA.

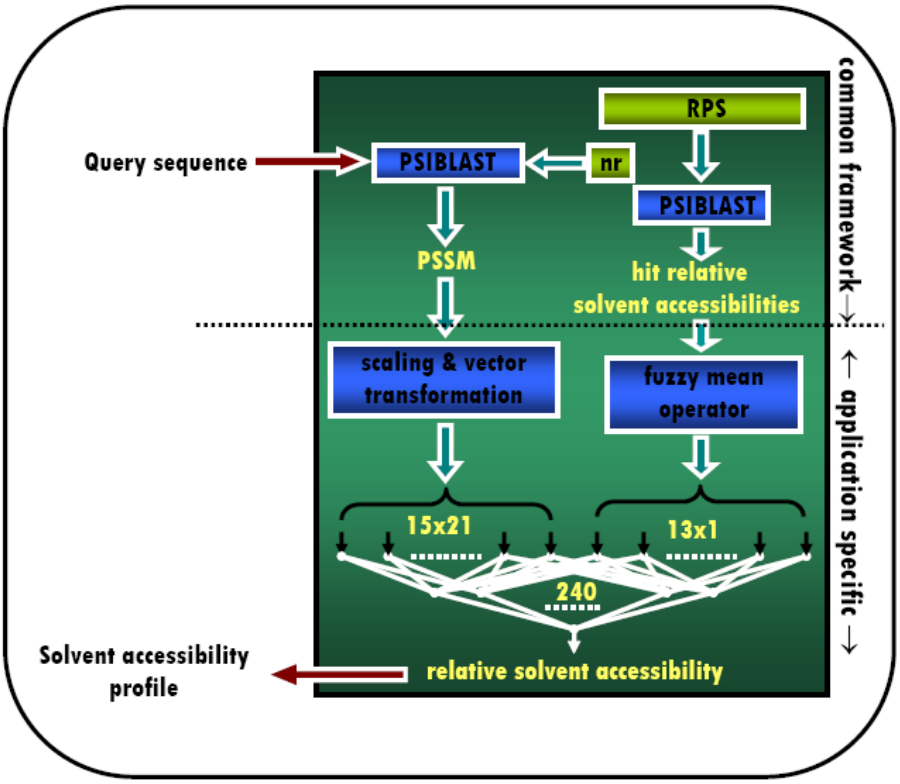


Figure 4.2: The block diagram of the MUPRED solvent accessibility prediction system. The profile of the query protein is first calculated and used to generate two feature sets. The first set consists of vectors derived from the normalized and rescaled PSSM using a sliding window scheme with window length (W) 15. This set consists of 15x21 =315 features/residue. The second feature set is generated by searching the local database of representative proteins by profile-sequence alignment. The compatible fragments returned by the search process are used to estimate the relative solvent accessibility of each residue using the fuzzy mean operator. The vectors representing the second feature set consist of 26 features, resulting in 341 features altogether, representing each residue. The neural network consists of 240 hidden units and a single output neuron that produces the predicted solvent accessibility. The components above the dotted line are part of our common framework while, the components below are specific to the solvent accessibility prediction application.

4.2.2 Databases

We use the RPS described in Chapter 2 to estimate the relative solvent accessibility of the query protein. We employed two widely used datasets (benchmark sets) to compare the performance of MUPRED with other methods. The first database was introduced by the Rost and Sander in the context of the protein secondary structure prediction [Rost and Sander, 1994]. It contains 126 representative proteins with 23,426 residues (hereafter referred as RS126). The second data set was introduced by Naderi-Manesh [Manesh et al, 2001] in the context of information theory based solvent accessibility prediction method. The database consists of 215 representative proteins with 51,939 residues (hereafter referred as MN215). The proteins in RPS that are similar to the proteins in the benchmark sets are eliminated using the procedure explained in Chapter 3 (i.e., each sequence in the RPS database was queried against proteins in the benchmark sets using the BLAST program. If a hit with an e-value less than 0.01 is found, the query sequence was eliminated from the RPS). This procedure further reduced the number of proteins in RPS to 1657. In addition to testing our method on the two benchmark sets, we employed a third dataset derived from the Astral SCOP domain database version 1.69. The proteins in Astral SCOP dataset that are similar to the proteins in the RPS are discarded using the same procedure outlined above. The same filtration criteria were used to filter the third benchmark data set. The remaining 3386 domain sequences with 636,693 residues after the filtering make up the independent benchmark set.

4.2.3 Method

In the PSSM, each residue is represented by a 21 dimensional vector representing the likelihood of each of the 20 amino acids in that position. The profiles are first normalized and then rescaled in to [-1 1] before converting them into vectors suitable for neural network training. We found that the maximum and minimum values in the profiles of all proteins in the RPS were -10 and 12, respectively. Therefore, the profiles were normalized and rescaled using Equation (4.2) below:

$$PSSM(i,j) \leftarrow 2x - 1, \text{ where } x \leftarrow \frac{(PSSM(i,j) + 10)}{22}, \tag{4.2}$$

where $i \in [1,\ldots,n]$, n is the length of the query protein and $j \in \{A,C,D,E,F,G,H,I,K,L,M,N,P,Q,R,S,T,V,W,Y\}$. To create the first feature set, the transformed PSSM values are converted into vectors suitable for neural network training using a sliding window scheme i.e., vector representing the current residue is flanked by the vectors representing the neighbors on the both sides. This scheme allows us to capture the idea that a particular residue's solvent accessibility is dependent on the solvent accessibility states of its neighbors [Manesh et al, 2001, Ahmed et al 2002]. The number of neighbors on each side is determined by parameter *W*. The termini information of the protein is encoded in the vector using an additional bit. We arbitrarily choose 1 to represent the ends of protein, while 0 is used for representing the interior of the protein. We experimentally determined that the optimal number of neighbors on each side of the current residue to consider for this feature set is 7 and therefore the total number of features in this set is 21x15=315.

Similar to the PSSM feature set, the fuzzy means are first rescaled and converted into vectors suitable for training the neural network using the sliding window scheme. Again, we use an extra bit to indicate the termini of the protein using the same encoding as the PSSM feature set. We experimentally determined that the optimal window size is 13 and therefore the total number of features in this feature set is 2x13=26. These two feature sets together (26+315 = 341 features/residue) are used to train the neural networks. The neural network to integrate the fuzzy means and PSSM is a fully connected feed-forward neural network with one hidden layer, trained using standard back-propagation learning. We trained the networks with different number of nodes, starting at 170 and increase 10 units at a time. We found that 240 nodes result in an optimal performance. The output layer consists of a single neuron that produces the predicted RSA. The neural network has the following architecture 341×240×1 (input nodes × hidden nodes × output node). We randomly selected 50 of RPS proteins for generating the validation vectors and used the rest for training the neural networks. The networks were trained until the performance using the validation vectors started to decline. A total of 100 networks were trained using random initialization and the top 6 networks were retained for the prediction purposes. Each of the query protein is simulated on all 6 networks and the average of the 6 networks is taken as the output of the prediction system.

4.3. *Results*

In this section, we first discuss the metrics to measure the performance of our prediction system and then discuss the performance of the fuzzy mean operator, fuzzy mean

operator followed by and neural network and finally, MUPRED that uses both fuzzy mean operator and PSSM on the RPS, independent SCOP derived set and the two benchmark sets. We then proceed to compare the performance of MUPRED with the performance of the existing methods on the two benchmarking datasets.

If the system is used as a classifier to group the residues into two classes (*buried* and *exposed*), the Q_2 and MCC were used to assess the performance:

$$\text{Accuracy}\,(Q_2) = \frac{p+n}{t}, \tag{4.3}$$

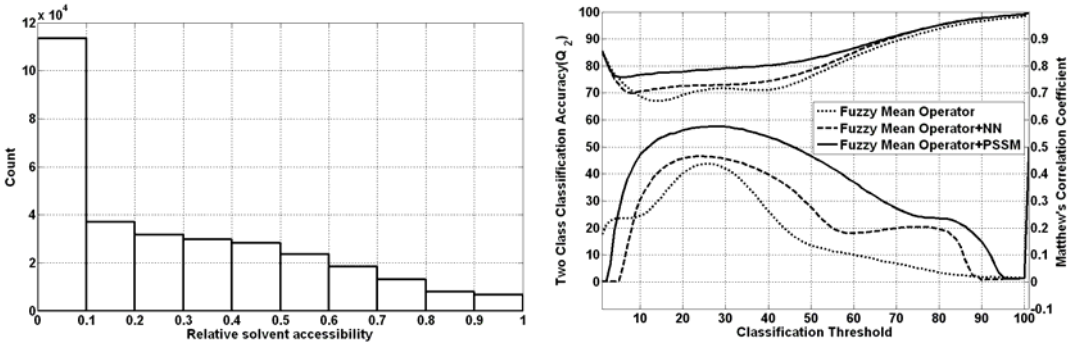
In order to assess the performance of the real value SA prediction ability of the system, the (MAE) as defined below is used:

$$MAE = \frac{1}{N}\sum \left| RSA_{observed} - RSA_{predicted} \right|, \tag{4.4}$$

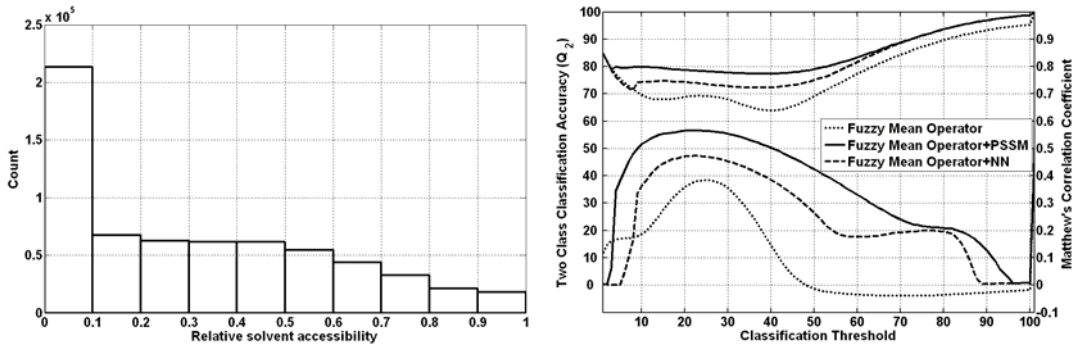
where $RSA_{observed}$ is the experimental RSA obtained by dividing the actual solvent accessibility (from DSSP files) by their respective maximum SA.

When we tested the SA profile generated by the fuzzy mean operator alone, we noticed that the trend of predicted SA profile resembles the actual SA profile, except that dynamic range of the predicted SA profile is consistently smaller. Since the neural networks function well as the signal amplifiers, we trained a neural network using the sliding window scheme described in Section 4.2.3 with the window size 13. This network was not used in the final MUPRED as both the features are being integrated

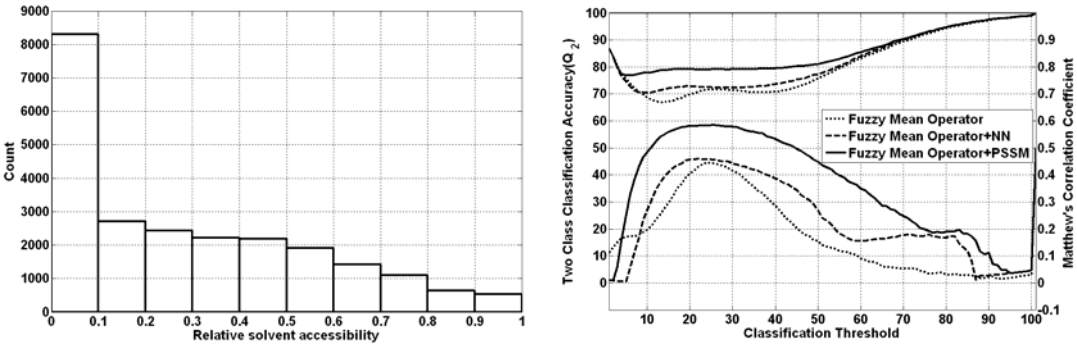
using a neural network, and there seems to be no practical advantage in amplifying signals before integrating [Bondugula and Xu, 2007]. The performances of our systems as a two class-classifiers on the various datasets are given in Figure 4.3 (a), 4.3(b), 4.3(c) and 4.3(d). The plot on the left illustrates the distribution of the RSA in the corresponding dataset, while the plot on the right contains the classification accuracies and the Matthew's correlation coefficients at various classification thresholds. In the plots depicting the accuracies, the two-class classification accuracy is plotted using the horizontal axis and vertical axis on the left while, the Matthew's correlation coefficient is plotted using the horizontal axis and vertical axis on the right. Note that the results in Figure 4.3 (a) are not re-substitution errors as the query protein is eliminated from RPS when its RSA is being predicted.



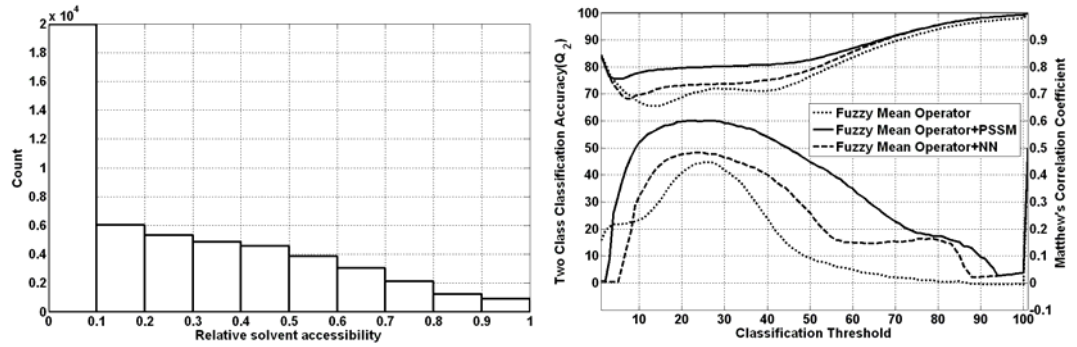
(a)



(b)



(c)



(d)

Figure 4.3: The histograms of the RSA in various data sets (Left). Performance of our methods on each of the data sets (Right). The classification threshold is varied along the horizontal axis, while the two-class classification accuracy is plotted using the vertical axis on the left while, the Matthew's correlation coefficient is plotted using the vertical axis on the right. (a) Training set of 1657 proteins, (b) SCOP data set with 3457 proteins, (c) Rost and Sander 126 protein set, and (d) Manesh 215 protein set.

We compare the performance of MUPRED to existing methods on the two most widely used sets. The comparison of existing methods with our method on the RS126 dataset is presented in Table 4.1, while the comparison on the MN215 dataset is presented in Table 4.2. We convert the RSA to various classes based on the thresholds used by the other methods. From Table 4.1, it can be noticed that the performance of MUPRED is slightly inferior to other methods at lower thresholds. We believe the small size of the database (only 126 proteins) played a role in this regard, as the same trend is not seen in Table 4.2. In fact, MUPRED has a higher accuracy than all other methods at all thresholds, with the exception at threshold 64%, where the methods based on information theory by Manesh *et al.*, [2001] has higher accuracy. Also, MUPRED has higher accuracy than other methods when three-class classification is used (last row in Table 4.1).

Table 4.1: The performance comparison of MUPRED with existing methods on the RS126 data set

Threshold/Method	**MUPRED**	**PHDAcc**	**IT**	**SVMPsi**	**FKNN**
0	87.1	86.0		86.2	87.2
5	76.8			79.8	82.2
9	77.9	74.6	78.2		
16	79.2	75.0	77.5	77.8	79.0
23	79.1		77.4		
25	79.2			76.8	78.3
9/36 (three class)	68.5	57.5	61.5	59.6	63.8

PHDAcc – Rost and Sander, 1994

IT- Manesh *et al.*, 2001

SVMPsi- Kim and Park, 2004

FKNN – *Sim et al.*, 2005

Table 4.2: The performance comparison of MUPRED with existing methods on the MN215 data set

Threshold/Method	MUPRED	IT	NETASA	SABLE	SARPred
4	76.9	75.1			
5	76.9		74.6	76.8	74.9
9	77.9	75.9			
10	78.1		71.2	77.5	77.2
16	79.2	75.5			
20	79.1			77.9	77.7
25	79.2	74.4	70.3	77.6	
30	79.3				77.8
36	79.6	74.1			
40	79.6				78.1
49	81.2	79.9			
50	81.5		75.9		80.5
60	86.2				85.3
64	87.8	97.2			
70	90.9				90.7
80	95.2				95.1
81	95.6	80.5			

NETASA- Ahmed and Gromiha, 2002

SABLE-Adamczak *et al.*, 2004

SARPred- Garg *et al.*, 2005

The MAEs of MUPRED on RPS, the SCOP derived independent set, RS126 and MN215 are 14.17, 15.29, 14.31 and 13.6, respectively. The Pearson correlation coefficients of our method on RPS, the SCOP derived independent set, RS126 and MN215 are 0.72, 0.69, 0.71 and 0.72, respectively. Garg et al. [2005] reported the Pearson correlation coefficient of 0.67 on the MN215 dataset. The MAE and the Pearson correlation coefficient on the RPS and the SCOP derived set indicate that the overtraining did not occur when we trained our neural networks.

The software for MUPRED SA prediction system is more than 98% similar to the MUPRED SSP program. In fact, we extended our MUPRED SSP program by integrating the SA prediction specific code and we now make combined prediction of secondary structure and solvent accessibility. The addition of SA specific code increases the prediction time only by a few hundredths of a second. So, our average prediction time of 0.55 sec/residue still holds.

4.4 Summary

We propose a novel SA prediction system that is similar to our secondary structure prediction system, the difference being, the former is a function approximation, while the later is a classification problem. The method is based on our framework for protein structure prediction. Our method uses the structural information in the PDB more efficiently than the existing methods and therefore, reduces the dependence on the homologous sequences in the databases for a well defined profile. Our system provides a balanced predicted for the sequences that have homologs in the database of proteins with known structures and for sequences that have no close homologs. Our results prove that the additional information provided by using the structural information has boosted the prediction accuracy considerably. In case of sequences with many homologs, the profile (PSSM) is well defined and to the prediction, along with the fuzzy means. In other cases where the sequence does not have close homologs, the system predicts the SA from the homologous fragments and hence the prediction is heavily influenced by the FMO. The system with fuzzy mean operator followed by a neural network, emulates an extreme case of MUPRED where there is no information from the PSSM and the prediction is solely

based on the homologous fragments. The results of this system help us estimate the lower bound of accuracy in such cases. One of the appealing features of our systems is that our system never needs to re-trained or re-tuned. As more and more representative structures are solved, their sequences just need to be added to the RPS and the algorithm will use the new information immediately. Using our system, in addition to obtaining the RSA, the user can multiply the RSA by their maximum solvent accessible areas of respective amino acids to obtain the real solvent accessibility values.

5. Application of Our Framework for Protein Tertiary Structure Prediction

5.1 Motivation

Traditionally, the X-Ray diffraction patterns or the spectral properties of the folded proteins are used to determine the structure of the proteins experimentally. These methods are often time consuming and expensive. Due to rapid advancements in the sequencing technologies, many new complete genomes are available each year, there by contributing more proteins that need to be structurally characterized. It will take many years and cost millions of dollars if we rely on experimental methods alone. The field of protein structure prediction has offered an alternative solution to this problem by computationally predicting the structure of a protein from its sequence. There are two popular computational methods for protein structure prediction. They are *Ab initio* [Li and Sheraga, 1987; Pedersen and Moult, 1997; Bradley *et al.*, 2005], homology modeling [Bowie and Eisenberg, 1991; Ring and Cohen 1993; Chivian and Baker, 2006]. *Ab initio* method is structure prediction from first principles by minimizing an energy function that includes the physical and statistical properties of amino acids. In homology modeling, the structural information from close homologs of the query protein is used to predict the structure of the query protein. Homology modeling works with the assumption that at least one close homolog of the query protein exists in the database of proteins with

known structures. *Ab initio* methods often demand huge computational resources and are seldom accurate.

5.2 Tertiary structure prediction

5.2.1 Main method

A hybrid approach called "mini-threading" [Simons et al., 1997; Bystroff and Baker, 1999; Inbar et al., 2003; Chikenji et al., 2003; Lee et al., 2004] has demonstrated great potential. Mini-threading obtains matches between a query sequence and short structure fragments in PDB for building local structures. Once local structures are more or less defined, assembling them results in a significantly smaller computational search space and a better chance to achieve high prediction accuracy. Some success of mini-threading has been demonstrated in the Critical Assessment of Techniques for Protein Structure Prediction (CASP) [Venclovas et al., 2001; Venclovas et al., 2003] and various examples [Li et al., 2004; Bradley et al., 2005].

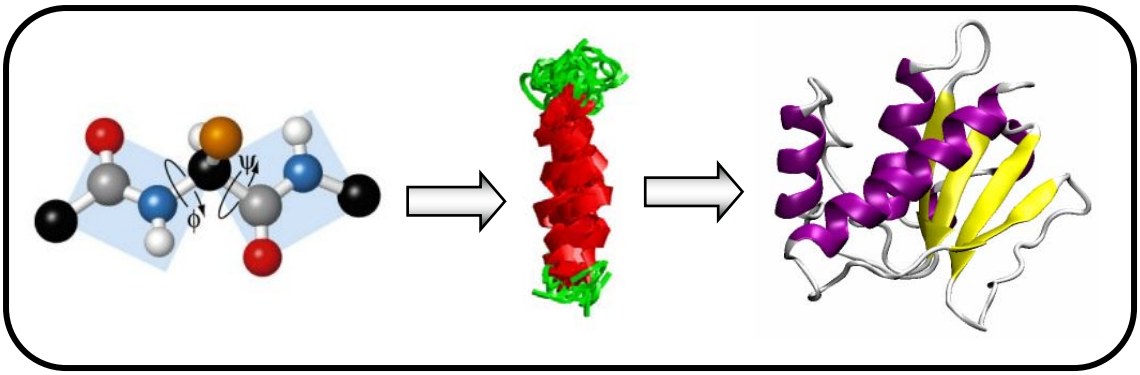


Figure 5.1: Illustration of mini-threading approach. Left: Φ and Ψ angles of protein backbone; middle: structure distribution based on the Φ/Ψangle distributions; right: final prediction of the structural model.

Figure 5.1 illustrates the basic process of existing mini-threading methods [Simons et al., 1997] following these four steps:

1. Search for compatible fragments of short sequences in a query protein against PDB. Typically, the query sequence is divided into 9 consecutive amino acids, or query 9-mers, for this purpose. The search criterion is based on sequence identity of gapless alignment between the query 9-mer and a 9-mer sequence of a known protein structure. The search generally yields a number of significant hits for each query 9-mer.
2. Build Phi-Psi angle distributions. The conformation of the protein backbone is defined by Phi-Psi angle on the protein backbone (see Figure 5.1(left)). Among the structural fragment hits for each query 9-mer, the Phi-Psi angle distributions may be so narrow in some cases that the structures of these hits are similar (see Figure

5.1(middle)). In other cases, the Phi-Psi angle distributions may be so wide that the structures of the hits are diverse, but still contain certain structural information.

3. Assemble the 9-mer fragments into a unified global structural model. Such an assembly can be done using genetic algorithms [Bystroff and Baker, 1999] or Monte Carlo simulations [Skolnick and Kolinski, 1991]. The distributions of the Phi-Psi angles are used as soft constraints (which will be referred to as restraints in this proposal), together with other energy functions such as a pairwise energy function between amino acids, for assembling the structure.
4. Group the generated structures into clusters. Clusters are ranked by their average energy functions or number of generated structures in the cluster. Typically, the best structure clusters is chosen as the final prediction (see Figure 5.1(right)).

Although highly promising, the mini-threading approach is still in its infancy. The existing mini-threading methods not only require long computing time but also fail to yield good results consistently. In addition, the prediction qualities of the exiting methods are user-specific, i.e., good performance is often coupled with extensive human intervention by experienced researchers. Few users other than the tool developers have successfully applied the mini-threading approach to their research. We think that the following limitations are the major barriers to achieve consistent performance in prediction accuracy and speed:

1. The selection of 9-mer fragments as mini-template is somewhat arbitrary. Although some tests have shown that 9-mer is better than other fixed length fragments, suitable division lengths vary among different proteins as well as different regions of an individual protein. Hence, fixed 9-mer fragments under-utilize the structural information in the database when longer fragments are available. In fact, it has been found that for a 15-residue query sequence fragment (15-mer) there is a 91% probability to find a matching fragment in PDB within 2 Å root mean square deviation (RMSD) [Du et al., 2003], and some even longer fragments can also be used for assembly [Jones and McGuffin, 2003]. Although a systematic method in identifying these fragments is still lacking, in principle longer fragments can be used for mini-threading. In addition, the chance to find longer structural fragment is increasing, as more and more protein structures are being solved.
2. The information used from mini-template is limited to independent angular restraints based on the Φ and Ψ angle distributions at the individual amino acid level. In many cases, when a small variation is introduced in the conformation of a fragment (such as the structural variation between the fragments of similar sequences in PDB), the restraints are created not only at the Φ and Ψ angles of each amino acid, but also at the correlations between angles of different amino acids. Such correlations on the restraints have not been used in mini-threading.
3. The heuristic optimization methods such as genetic algorithms and Monte Carlo simulations often generate protein structures that are far from the global optimal solution of an energy function. Typically, many runs with different initial conditions

have to be performed hoping to capture structural conformations that are close enough to the true structure. Therefore, the computation is very time consuming and it often requires supercomputers or computer clusters to achieve results in hours or days.

We propose a novel method based on the idea of mini-threading that overcomes the disadvantages discussed above. The following are the differences between our method and the mini-threading method and each of these differences overcomes one disadvantage discussed above. First, we use the fragments generated by our framework. This procedure is fast and results in similar fragments of various lengths to the query protein, based on the local sequence alignments. Second, we use the Cartesian coordinates of the amino acids of the similar fragments, instead of dihedral angles. Since we use the predicted secondary structure as guidance we already incorporate the dihedral angle information. By using Cartesian coordinates together with distance constraints between amino acids, our method can utilize the correlational information among dihedral angles more effectively than other mini-threading methods. Third, we use the multidimensional scaling (MDS) [Borg and Groenen, 1997] method to deduce the structure from the compatible fragments. MDS method is extremely fast and computationally efficient, when compared to other traditional methods.

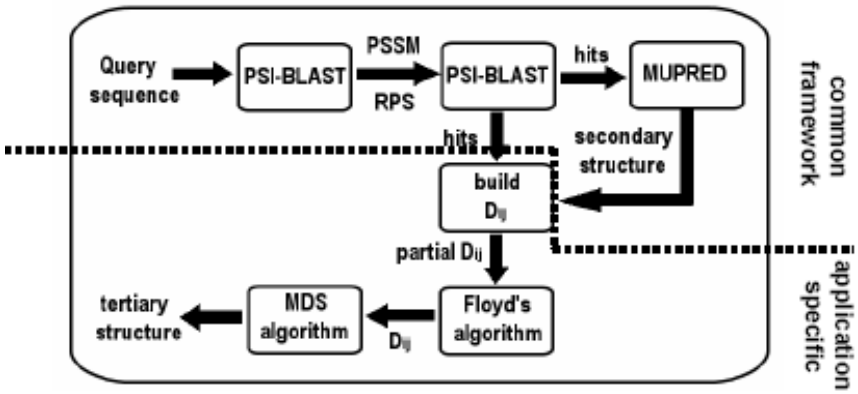


Figure 5.2: The block diagram of our method for tertiary structure prediction. First, the PSSM of the query protein is generated using the PSI-BLAST program. The PSSM is used to perform profile-sequence alignments with the proteins in the representative protein set. The compatible fragments are used to predict the secondary structure of the query protein using MUPRED program. The secondary structure of the query protein, along with the compatible fragments is used to build a pair-wise distance matrix. The unfilled elements in the partially filled distance matrix are extrapolated using the Floyd's shortest-path algorithm [Floyd, 1962]. The full distance matrix is passed through the multi-dimensional scaling algorithm to convert the distances in to Cartesian coordinates of the C-alpha atoms of the predicted structure. The components above the dotted line are provided by the common framework, while the components below are specific to the protein tertiary structure prediction.

5.2.2 Databases

Our framework uses the RPS described in Chapter 2, for searching the compatible fragments. In order to test our method, we created a test protein database (TDB) from the AstralSCOP database. The 25% identity filtered Astral SCOP version 1.69 database consists of 5457 protein domains. To prevent predicting the structures of proteins that are similar to the proteins in our RPS (trivial cases), we used the following procedure to add the proteins to TDB: each sequence in the Astral SCOP database was queried against RPS using the BLAST program. If no hit with an e-value less than 0.01 was found, the

sequence was added to the TDB. This process resulted in a TDB of 1304 protein domain sequences. Out of these 1304, we randomly selected 200 proteins to test our method.

5.2.4 Scoring Function

The compatible fragments produced by our framework are scored with the following scheme:

$$S = 5 - \log_{10}(Evalue) \tag{5.1}$$

The score '*S*' is a 'similarity' measure that is meant to differentiate between the hits of different statistical significance. It is designed such that the hit with least statistical significance (E-value = 11,000) should also receive a positive score. Other alternative similarity measure that yields a higher score for statistically significant hits, lower score for statistically insignificant hits and positive scores for all possible hits may be used in place of Equation (5.1). Note that only the relative scores, not their absolute values are important.

5.3 Problem Formulation

In this work, we formulate the problem of protein tertiary structure prediction as a graph realization problem. Assume that there are *n* points $x_i \in R^3, i = 1,...,n,$ in a 3D space, each point representing the C-alpha coordinates of a protein with *n* amino acids. Let $D=(d_{ij})$ represent a distance restraint matrix such that the element d_{ij} represents the Euclidean distance between amino acid *i* and *j*. Suppose, we know the values of some edges from

the fragments that are aligned with the query protein, then the graph realization problem is to determine the coordinates of the points from the partial distance restraint matrices such that the Euclidean distance between each pair of points match the given distance restraints, $\left\| X_i - X_j \right\| = d_{ij}$ for all available d_{ij}. Using the fragments obtained under our framework, some pairs are over-restrained (multiple hits that have a residues aligned at positions *i* and *j*) and some pairs are under-restrained (no hits have residues aligned at positions *i* and *j*). Also, when the distance restraints are inaccurate estimations, usually there is no exact or unique solution to the over-determined system of equations. Instead, the problem is formulated as an optimization problem that minimizes the sum of squared errors. The basic realization problem can be formulated as the squared error function:

$$\min_{X_1 \ldots X_n \in R^3} \sum_{i,j=1,\ldots,n} \left(\left\| X_i - X_j \right\| - d_{ij} \right)^2, \tag{5.2}$$

Since the fragments returned by our framework have associated E-value, the score S can be used as the relative importance of various hits. The revised objective function that uses the score is presented in Equation (5.3):

$$\min_{X_1 \ldots X_n \in R^3} \sum_{i,j=1,\ldots,n} S \left(\left\| X_i - X_j \right\| - d_{ij} \right)^2, \tag{5.3}$$

The above formulation allows us to use the compatible fragments any size, as opposed to fixed size fragments such as 9-mers used by other mini-threading methods. The formulation also facilitates the optimization on Cartesian coordinates instead of dihedral angles, using the information in PDB more efficiently. The efficiency comes from the fact that Cartesian coordinates capture the restraints between the amino acids.

5.3.1 Multi-dimensional Scaling (MDS)

The error surfaces of the objective functions in Equations (5.2-5.3) are generally non-convex with many local minima. Local and global optimizations using traditional techniques have many disadvantages. For example, the success of local optimization techniques like Levenberg-Marquardt [Levenberg, 1944; Marquardt, 1963] depend on good initial points while, the global optimization techniques like simulated annealing or genetic algorithms are very slow in large continuous search spaces and demand huge computational resources.

We found that MDS, related to principal component analysis, factor analysis, and cluster analysis, is suitable for our optimization problem. The MDS method is an efficient method for solving the graph realization problem. MDS is a set of data analysis techniques that display the structure of distance-like data as a geometrical picture. MDS starts with one or more distance matrices (or dissimilarity matrices) that are presumed to have been derived from points in a multidimensional space, and it finds a placement of the points in a low-dimensional space, where the distances between points resemble the original dissimilarities. In our optimization problem, the restraints between the residues form the dissimilarity matrix. We started with the classical metric MDS, the simplest and most efficient MDS algorithm. In classical metric MDS, the data is quantitative and the proximities of objects are treated as distances in the Euclidean space. The goal of metric MDS is to find a configuration of points in a multidimensional space such that the inter-point distances are related to the provided proximities by some transformation (e.g., a linear transformation). If the proximity data were measured without error in the

Euclidean space, then classical metric MDS would exactly recreate the configuration of points. In practice, the technique tolerates error gracefully, due to the over-determined nature of the solution. This will be very helpful when we apply it to our new protein structure prediction problem formulation, as our dissimilarity data can be very inaccurate and inconsistent (over-restrained). Classical metric MDS (CMDS) is suitable for high-dimensional problems, since its main operation is singular value decomposition on a matrix with the same size of the dissimilarity matrix, where efficient algorithms exist. This means that the CMDS can work well for our protein structure prediction problem with a large number of residues.

In CMDS, first a double centered matrix B is calculated from D using the following equation:

$$B = -\frac{1}{2}\left(d_{ij}^2 - \frac{1}{n}\sum_{i=1}^{n} d_{ij}^2 - \frac{1}{n}\sum_{j=1}^{n} d_{ij}^2 + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}^2 \right) = X_i' X_j \quad for \;\; i, j = 1,...,n, \qquad (5.4)$$

The matrix B is decomposed into $B = UVU'$ using singular value decomposition or Eigen decomposition, leading to coordinate solution $X = UV^{1/2}$. In our case, the distance restraints are approximate and inaccurate, there by the B matrix can be indefinite with negative as well as zero or positive roots. In such cases, V is a high-dimensional matrix. We choose the three Eigen vectors U_3 that correspond to the three largest Eigen values of V, represented as V_3. If $Y=U_3V_3^{1/2}$, then $C = Y'Y$ is a least squares approximation to B.

While the objective function in Equation (5.2) is easy to handle in CMDS, the weighted objective function in Equation (5.3) is difficult to handle in CMDS, but is

straightforward using weighted MDS (WMDS), the drawback is that it is computationally much more expensive than the CMDS.

5.3.2 Structure Prediction Algorithm

We use the following algorithm to predict the three-dimensional structure of the query protein.

Begin

1. Read the compatible fragment information, secondary structure information and Cartesian coordinate information generated by our framework.
2. Using the information in step 1, predict the secondary structure of the query protein using MUPRED or any other SSP algorithm.
3. Initialize the distance matrix to 0 for the elements on the diagonal and to high values elsewhere.
4. *Repeat until* the number of unmarked segments is 0
    a. Randomly select a secondary structure segment and mark the segment as processed
    b. Find all the hit fragments that have a similar secondary structure segment (defined by the SOV [Zemla et al., 1999] cutoff) aligned with current segment.
    c. Select one of the segments with probability linearly proportional to its score *S*.

d. Scan the hit on either side to see if the hit also has a match for other secondary structure segments. If there is match, mark the segment and extend the boundary of the selected hit

e. Copy the pair-wise distances to the D_{ij} distance matrix

End Repeat

5. Extrapolate the remaining elements in the distance matrix using Floyd's shortest path algorithm.

6. Transform the distance information into configuration of points using multi-dimensional scaling algorithm. The coordinates of three dimensions with highest Eigen values are the predicted C-alpha coordinates of the query protein.

End

At first, it might seem that the prediction accuracy of our method is highly dependent on SSP accuracy. In fact, this is not the case. When we select the possible hit fragments, we use the SOV cutoff of 50. That is, if at least half of the secondary segment is correctly predicted by the SSP algorithm, there is possibility that it will be used by the algorithm for the template information. In our implementation, the structure information can be used even if a secondary structure segment is totally missed by the SSP algorithm. This is especially true if the two end secondary structure segments are predicted with at least 50% accuracy, suppose if one of them is currently selected, all the remaining segments are checked if they can contribute to the prediction. If the first and the last segments match, the entire hit is used even if the segment in the middle was missed by

the SSP algorithm. The only factor that influences the accuracy of our algorithm is the ability of the SSP algorithm to correctly identify as many secondary structure segments as possible, even if they are predicted with incorrect boundaries. The selection of hit fragments to get the template information is illustrated in Figure 5.3.

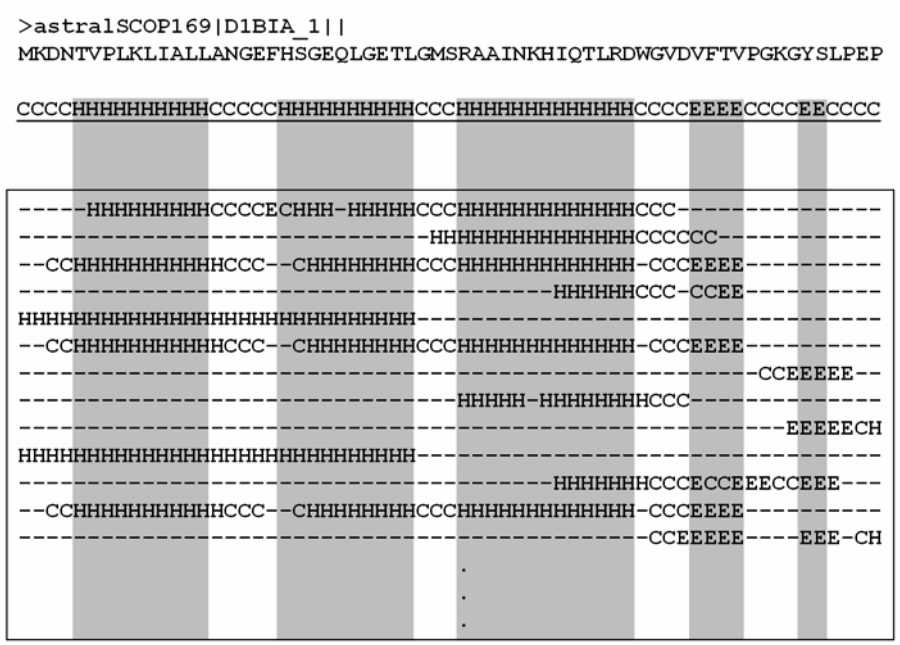


Figure 5.3: The selection process illustrated. Top- query protein. Middle- predicted secondary structure. Bottom- the secondary structures of the database hits. For each predicted secondary structure segments (shaded), one of the database fragments is chosen based on its score.

During the processing of building the D matrix, some pairs of residues are not filled. There are two possible reasons for the unfilled elements. First, for a residue pair (*i*,*j*), there may not be a hit that has residue aligned at both positions *i* and *j*. Second, even if there was a hit that has residues aligned at both position *i* and *j*, it may not be selected due to randomness involved in the selection of hits for the distance information. If the

position (*i*,*j*) is not filled, we extrapolate the distance using the efficient all-pairs shortest-path algorithms, such as Floyd's shortest path algorithms.

In our earlier work [Bondugula *et al.*, 2006], we used the same problem formulation and the CMDS algorithm. This version of tertiary structure prediction algorithm is more basic, i.e., it neither uses the predicted secondary structures, nor the scores during MDS algorithm. Here, for every possible pair of residues (i,j, i≠j), we search for hits that have residues aligned with current residues i and j. We choose one of the hit fragments to fill d_{ij} with a probability that is proportional to the value of the score *S* associated with each hit. Similar to our more advanced version described above, the unfilled positions are extrapolated using Floyd's shortest path algorithm.

5.3 Results

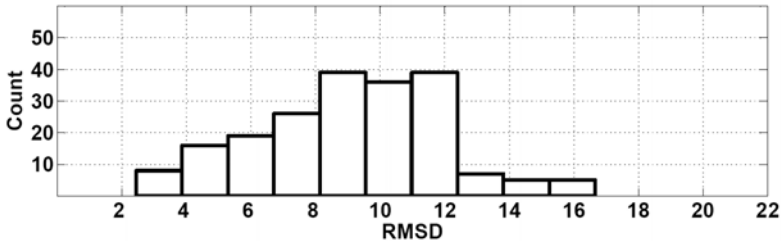
For each protein in the TDB, we used our algorithm to generate several thousand structures using both CMDS and WMDS. Currently, we are still developing methods to automatically select the best structure yet. In this work, we use the native structure as a guide to pick the best predicted structure. Using native structure to select the best of the generated structures will help us estimate the upper bound of our algorithm. We compare each of generated structure with the native structure and retain the one with the lowest RMSD as the best predicted structure. We first generated 2,000 structures per protein using CMDS algorithm. Next, we repeated the experiment by changing the reconstruction procedure from CMDS to WMDS. We then increased the number of structures to 10,000

per protein. We compare the results of the above three variants in this work with the performance of our earlier approach without the secondary structure information.

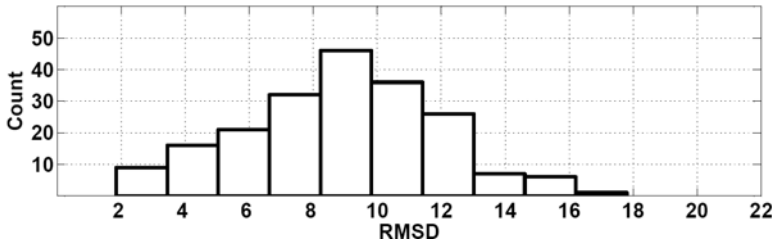
In Figure 5.4, we present the results of variants of our approach on the 200 test protein sequences in the TDB. The histogram of the results of the current approach using CMDS and the best of 2000 structures are presented in Figure 5.4 (a). In Figure 5.4 (b), we changed CMDS to WMDS and noticed that in 68% of the proteins the RMSD was reduced, in the remaining 32%, the RMSD was slightly increased. The RMSD distribution is presented in Figure 3 (b). We then proceeded to increase the number of structures generated from 2,000 to 10,000, still using the WMDS method. We noticed that the RMSD drops in 88% of the proteins when compared to the previous variation. The distribution of RMSD for this variant is presented in Figure 5.4 (c). In Figure 5.4 (d), we present the histogram of RMSD distribution using our previous approach that does not use secondary structure, uses CMDS and selects the best of 2000 structures. Overlap of the RMSD distributions is illustrated in Figure 5.5. It can be noticed that the variation that uses WMDS and 10,000 structures has more structures in the lower RMSD (left) region.

The contribution of the predicted secondary structure input into the tertiary structure prediction method can be estimated by comparing the results of our previous approach with the first variant in the current work. They both use CMDS method and select the best structure among the 2000 predicted structures/protein. When we compared

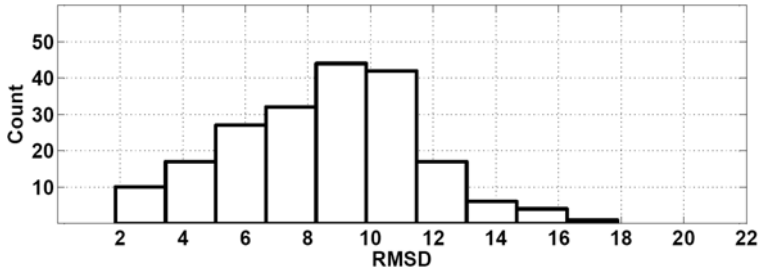
their results, we found that inclusion of the predicted structure help reduction of RMSD in 84% of the proteins. When compared our previous version's performance with our third variation (WMDS and 10,000 structures), we noticed that the RMSD drops in 92% of the proteins. The drop in RMSD is illustrated in Figure 5.6. The x-coordinate is the RMSD using the previous method (no secondary structure, CMDS and 2,000 structures) and the y-coordinate is the RMSD using the work presented in our third variation (WMDS with the best of 10,000 structures). Some statistics on the results are presented in Table 5.1. Three predictions that are in our top ten predictions are illustrated in Figure 5.7.



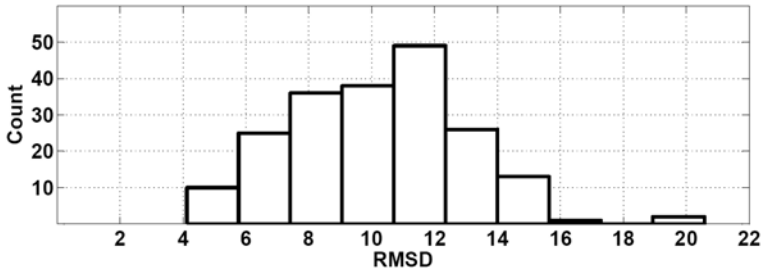
(a)



(b)



(c)



(d)

Figure 5.4: The histograms showing the distribution of RMSDs of the 200 proteins using various methods. (a) Secondary structure information and classical multi-dimensional scaling with selection of best of 2000 structures (b) Secondary structure information and weighted multi-dimensional scaling with selection of best of 2000 structures (c) Secondary structure information and weighted multi-dimensional scaling with selection of best of 10,000 structures (d) no secondary structure information and classical multi-dimensional scaling with selection of best of 2000 structures.

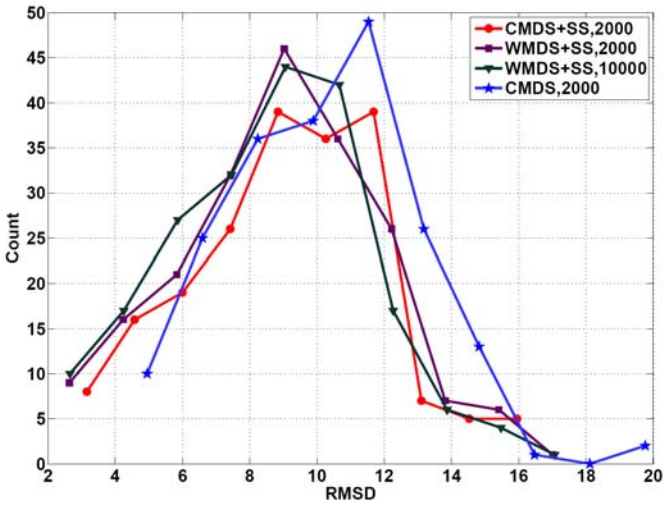


Figure 5.5: The histogram comparison of the four variants of the tertiary structure prediction program

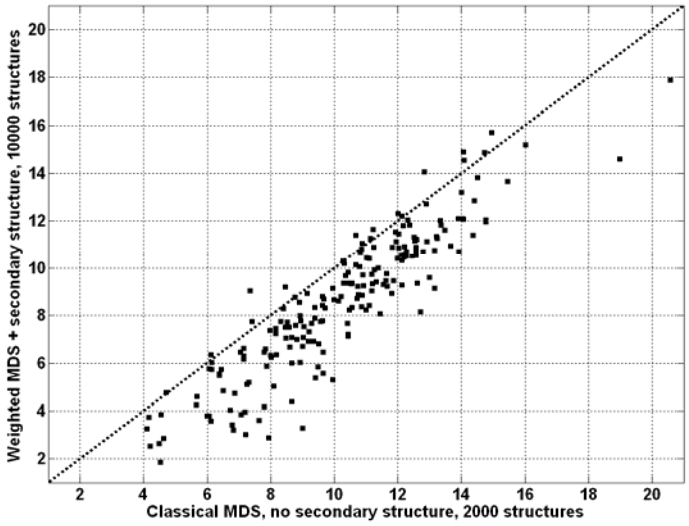


Figure 5.6: Performance comparison of our current method with our previous method on the 200 proteins in the test database. Each point represents a protein. The horizontal coordinate is the RMSD using the previous method (no secondary structure, classical multi-dimensional scaling and 2000 structures) and the vertical coordinate is the RMSD using the work presented in the current work (secondary structure + weighted multi-dimensional scaling with the best of 10000 structures). Reduction in the RMSD was noticed in 92% of the proteins using the current method.

Table 5.1: Statistics on performance of different variations

Description	**A**	**B**	**C**	**D**
Lowest RMSD	2.44	1.88	4.12	1.85
Cases in which RMSD < 4	4.5%	7.5%	0%	9%
Cases in which RMSD < 6	15%	17.5%	5%	20%

A- Classical Multidimensional scaling (with secondary structure), 2,000 structures

B- Weighted Multidimensional scaling (with secondary structure), 2,000 structures

C- Classical Multidimensional scaling (no secondary structure), 2,000 structures

D- Weighted Multidimensional scaling (with secondary structure), 10,000 structures

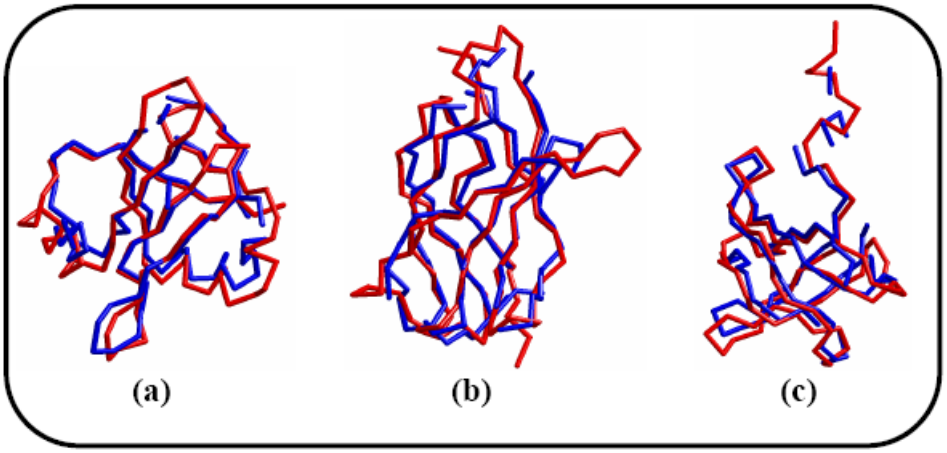


Figure 5.7: Some of our best prediction results. Red- native C-alpha trace derived from the crystal structure. Blue- predicted C-alpha trace using our method. (a) Human T-cell transcription factor NFAT1 (PDB code 1p7h) (b) E-Coli Ribosomal protein L25 (PDB code 1dfu) (c) Arthrobacter globiformis N, N-dimethylglycine oxidase, C-terminal domain (PDB code 1pj5)

Our framework and MUPRED SSP algorithm have linear time complexity. The process for building D also has a linear time complexity. The shortest-path algorithms have a complexity of $O(n^3)$, where n is the number of residues. The core of classical MDS is SVD, which has complexity $O(n^3)$. Therefore, our approach has a time complexity of $O(n^3)$. As a example, a query protein of length 96 residues, it took an average of 1.65 second to generate one structure using the CMDS and 5.13 second to generate one structure using WMDS on a Intel Xeon, 3.0 GHz processor. The memory requirement for the entire program is less than 50 megabytes of RAM. The times and memory requirements include the SSP of the query protein. Since we use the native structure of the protein to guide our selection process, it is not fair to compare our results with other methods.

5.4 Summary

We introduce a novel method for protein tertiary structure prediction that is formulated as a graph realization problem. Our framework and our problem formulation along with our objective function allow us to use the information in the PDB more efficiently than existing methods. Our method also uses computationally efficient MDS algorithms to transform the distance information into Cartesian coordinates. We show that incorporating the predicted secondary structure information into the tertiary structure prediction will boost the accuracy considerably. The accuracy of the secondary structure prediction has very little influence on the tertiary structure prediction accuracy. The structure of each protein can be predicted in few hours on a personal desktop computer that has a single processor. With very little effort, our program can modified to run on multiple processors to further reduce the computation time.

6. Conclusions and Future Work

6.1 Conclusions

This dissertation work was an attempt to push the state-of-the-art a little further for the problem of protein tertiary structure prediction. To address this problem we developed a novel framework. The framework facilitates new algorithms that bridge the gap between the methods that rely on only one type of information. In the context of secondary structure prediction, the framework helped in bridging the gap between template-based methods and the profile-based methods. In the context of the solvent accessibility, the framework opened a new avenue by facilitating the development of a new method that uses structural information directly. Both of these applications that were developed on top the framework have achieved the state-of-the-art in their respective areas. In the realm of tertiary structure prediction, the method developed based on this framework helped to bridge the gap between *ab initio* protein structure prediction and the homology modeling. Our formulation of the tertiary structure problem as a graph realization problem, our choice of objective function and the method to solve the optimization have resulted in a potentially very accurate method.

The notable feature of our framework and all the applications that are built on top of it is that, the software does not require re-training or re-tuning as the new representative protein structures are solved. Whenever new sequences are available, the relevant information needs to be appended to our database. The framework and

consequently all its applications immediately start using the newly added sequences. The complete potential of the framework can be realized when we can make sure that there is representative sequence for each class of protein, a representative structure for each possible fold and a representative solvent accessibility profile for each possible family of proteins. This condition ensures that the framework will provide all the information possible to applications that predict various structural aspects of proteins. Under these circumstances, the performance of the applications is only dependent on the algorithm used and not the available information.

6.2 Limitations

6.2.1 NMR structures are not used (redundant structures not used…)

We currently use only sequences whose structures were determined experimentally by X-Ray crystallography method in our RPS. If the sequences of the proteins whose structures were solved using NMR methods can be used, the size of the RPS would have been one and half times the current size of database, making a better use of the available structural information.

6.2.2 Tertiary structure prediction method cannot automatically select the best prediction

We cannot yet make a fair comparison of our tertiary structure prediction method's performance with other methods as we do not yet have a method to automatically select the best predicted structure. This limitation is also shared by many other tertiary structure prediction methods. We can only deduce the upper bound of the tertiary structure

prediction ability with our current method. Our collaborators on this project, Dr. Yi Shang's laboratory and Dr. Kosztin's laboratory are currently working on various methods to automatically select top structures and to enhance the predicted structures. Dr Shang's group is trying to use decision-tree analysis, packing functions and empirical energy functions for the selection process. Dr. Kosztin's group is mainly focusing on enhancing the predicted structure using optimization techniques.

6.3 Ideas for future work

6.3.1 Framework for Contact Map Prediction

Two residues are said to be in contact if they are separated by a distance less than a predefined threshold. A matrix that records these contacts for all possible pairs of residues in a given protein sequence is called a contact map. A contact map provides useful information about non-local contacts that help proteins form and maintain stable structures. While a contact map does not contain all information about the protein, it can be viewed as good two-dimensional representations of protein three-dimensional structure. A contact map for a protein of length l is a binary matrix of dimension lxl such that each element (i,j) is equal to 1 if the Euclidean distance between residues is less than a pre-defined threshold T, or equal to 0, otherwise. Contact maps are useful in protein three-dimensional prediction and for protein structure comparison [Carr et al., 2002; Caprara et al., 2004].

SSP methods mainly classify the residues into various classes based on the patterns in a small neighborhood (not totally true in case of β-strands). So is the case with

SA. On the other extreme, TSP methods, like ours, perform exceptionally well if long homologs can be identified (either using sequence homology, structure homology, SA profile similarity or their combination). Contact map methods can perform relatively better if homologous fragments of intermediate lengths can be identified. We demonstrated in Chapter 2 that our framework identifies fragments over a wide range of lengths. In fact, this was proved in MUPRED SSP application. Our results indicate the MUPRED gains significant overall accuracy by predicting β-strands (that form non-local contacts) better than the competing methods. We think the ability of our framework to identify fragments of intermediate and long homologs is the main reason. The same advantage may be used by the contact map predictions based on our framework.

6.3.2 Stand alone packages for our framework, MUPRED SSP and MUPRED SA

Making stand alone package of our framework may prompt other researchers to develop novel applications or use it for a purpose which we never anticipated. Stand-alone packages, both command line interface and the graphical user interface could be developed and distributed over the internet. Most of the popular methods for SSP and SA prediction have unfriendly command line interface, especially to biologists, for whom most of these tools are developed. So, a single GUI (graphical user interface) that includes our framework, MUPRED SSP and MUPRED SA prediction could be developed. These tools can be made available for download through the MUPRED website.

6.3.4 Ideas for improvement of our Tertiary Structure Prediction

Two of our applications, MUPRED SSP and MUPRED SAP currently represent the state-of-the-art in these areas. Both of their prediction accuracies stand at little more than 80%. The predicted secondary structure can be used as a query to identify structural homologs and the predicted solvent accessibilities could be used to identify proteins in the RPS that have similar environment. Using these features will enable applications to use homologs that cannot be readily identified by using query amino acid sequence alone and could improve the performance greatly.

Our framework is very efficient. As discussed before, more then 95% of time required by our framework is utilized by the PSI-BLAST program and processing the PSI-BLAST output and parsing DSSP files for structural information takes at most three seconds. Similarly, our SSP and SAP methods require a small fraction of a second in addition to time required for our frame work. However, the same cannot be said about our TSP program. The reason is that, the structures in the ensemble are generated in serial fashion. With little effort and expertise in parallel programming, the software can be parallelized, there by making efficient use of, now prevalent, multi-core/hyper-threaded/multi-processor/cluster systems.

Reference

1. Adamczak R, Porollo A, Meller J. Accurate prediction of solvent accessibility using neural networks-based regression. Proteins. 2004 Sep 1; 56(4):753-67.

2. Ahmad S, Gromiha MM. NETASA: neural network based prediction of solvent accessibility. Bioinformatics. 2002 Jun;18(6):819-24.

3. Ahmad S, Gromiha MM, Sarai A. Analysis and prediction of DNA-binding proteins and their binding residues based on composition, sequence and structural information. Bioinformatics. 2004 Mar 1;20(4):477-86. Epub 2004 Jan 22.

4. Aloy P, Pichaud M, Russell RB. Protein complexes: structure prediction challenges for the 21st century. Curr Opin Struct Biol. 2005; 15(1):15-22

5. Altschul, S.F., Gish, W., Miller, W., Myers, E.W. & Lipman, D.J. (1990) "Basic local alignment search tool." J. Mol. Biol. 215:403-410

6. Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 1997; 25:3389-3402.

7. Anfinsen CB, Haber E, Sela M, White FH Jr. The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain. Proc Natl Acad Sci U S A. 1961; 15(47):1309-14.

8. Baker, D. (2006). Prediction and design of macromolecular structures and interactions Philos Trans R Soc Lond B Biol Sci 361, 459-463

9. Baldi P, Brunak S, Frasconi P, Pollastri G, Soda G. Exploiting the Past and the Future in Protein Secondary Structure Prediction. Bioinformatics 1999; 15: 937-946.

10. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. The Protein Data Bank. Nucleic Acids Res 2000;28:235-242.

11. Bondugula R, Duzlevski O, Xu D. Profiles and Fuzzy K-Nearest neighbor Algorithm for Protein Secondary Structure Prediction. In Proceedings of 3rd Asia-Pacific Bioinformatics Conference, Singapore. London: Imperial College, 2005.

12. Bondugula R, Xu D, Shang Y. A Fast Algorithm for Low-Resolution Protein Structure Prediction. Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE, Vol., Iss., Aug. 2006 Pages:5826-5829

13. Bondugula R, Xu D. MUPRED: A Tool for Bridging the Gap between Template Based Methods and Sequence Profile Based Methods for Protein Secondary Structure Prediction. Proteins: 2007;66(3):664-70.

14. Borg I, Groenen P (1997). Modern Multidimensional Scaling, Theory and Applications. Springer-Verlag, New York.

15. Bowie JU, Luthy R, Eisenberg D (1991). A method to identify protein sequences that fold into a known three-dimensional structure. Science. 253:164-170.

16. Bradley P, Misura KM, Baker D (2005). Toward high-resolution de novo structure prediction for small proteins. Science. 309(5742):1868-71.

17. Brenner SE, Koehl P, Levitt M. The ASTRAL compendium for sequence and structure analysis. Nucleic Acids Research 2000; 28:254-256.

18. Bystroff C, Baker D (1999). Prediction of local structure in proteins using a library of sequence-structure motifs. J. Mol. Biol. 281:565-577.

19. Caprara A, Carr R, Istrail S, Lancia G, Walenz B. 1001 optimal pdb structure alignments: integer programming methods for finding the maximum contact map overlap, J.Comput.Biol. 2004 ;11(1):27–52.

20. Carr B, Hart W, Krasnogor N, Burke E, Hirst J, Smith J. Alignment of protein structures with a memetic evolutionary algorithm, in:GECCO-2002:Proc.Genetic and Evolutionary Comput. Conf., Morgan Kaufman, Los Altos, CA, 2002.

21. Chandonia JM, Karplus M. Neural networks for secondary structure and structural class predictions, Protein Science 1995;4(2):275-285.

22. Chen H, Zhou HX. Prediction of interface residues in protein-protein complexes by a consensus neural network method: test against NMR data. Proteins. 2005 Oct 1;61(1):21-35.

23. Chikenji G, Fujitsuka Y, Takada S (2003). A reversible fragment assembly method for de novo protein structure prediction. Journal of Chemical Physics. 119:6895-6903.

24. Chivian, D., Baker, D. (2006). Homology modeling using parametric alignment ensemble generation with consensus and energy-based model selection. Nucleic Acids Res.34(17):e112.

25. Chothia, C. The nature of the accessible and buried surfaces in proteins. J Molec. Biol. 1976; 105, 1−14

26. David MP, Asprer JJ, Ibana JS, Concepcion GP, Padlan EA. A study of the structural correlates of affinity maturation: antibody affinity as a function of chemical interactions, structural plasticity and stability. Mol Immunol. 2007 Feb;44(6):1342-51.

27. Dayhoff MO, Schwartz RM, Orcutt BC. A model of evolutionary change in proteins. (1978) In "Atlas of Protein Sequence and Structure" 5(3) M.O. Dayhoff (ed.), 345 - 352, National Biomedical Research Foundation, Washington.

28. Du P, Andrec M, Levy RM (2003). Have we seen all structures corresponding to short protein fragments in the Protein Data Bank? An update. Protein Eng. 16(6):407-414.

29. Dunbrack RL. Sequence comparison and protein structure prediction. Curr Opin Struct Biol. 2006; 16(3):374-384.

30. Eyal E, Najmanovich R, McConkey BJ, Edelman M, Sobolev V. Importance of solvent accessibility and contact surfaces in modeling side-chain conformations in proteins. J Comput Chem. 2004 Apr 15;25(5):712-24.

31. Floyd RW. Algorithm 97: Shortest path. *Commun. ACM* 1962;5(6): 345.

32. Garg A, Kaur H, Raghava G. Real value prediction of solvent accessibility in proteins using multiple sequence alignment and secondary structure. Proteins 2005; 61:318–324.

33. Geourjon C and Deleage G. SOPM: A self-optimized method for protein secondary structure prediction. Protein Eng 1994;7:157-164.

34. Ginalski K, Grishin NV, Godzik A, Rychlewski L (2005). Practical lessons from protein structure prediction. Nucleic Acids Res. 33(6):1874-91

35. Ginalski K. Comparative modeling for protein structure prediction. Curr Opin Struct Biol. 2006; 16(2):172-177.

36. Gromiha MM, Suwa M. Variation of amino acid properties in all-beta globular and outer membrane protein structures. Int J Biol Macromol. 2003 Sep;32(3-5):93-8.

37. Henikoff S, Henikoff J. Amino acid substitution matrices from protein blocks. Proc. Natl. Acad. Sci. USA. 89(biochemistry) 1992; 10915 - 10919.

38. Hobohm U, Sander C. Enlarged representative set of protein structures. Protein Science 1994;3:522.

39. Holbrook SR, Muskal SM, Kim SH. Predicting surface exposure of amino acids from protein sequence. Protein Eng 1990;3: 659-665

40. Holley LH and Karplus M. Protein secondary structure prediction with a neural network. Proc. Natl. Acad. Sci. USA 1989, 86:152-156.

41. Hoskins J, Lovell S, Blundell TL. An algorithm for predicting protein-protein interaction sites: Abnormally exposed amino acid residues and secondary structure elements. Protein Sci. 2006 May;15(5):1017-29

42. Huang LT, Saraboji K, Ho SY, Hwang SF, Ponnuswamy MN, Gromiha MM. Prediction of protein mutant stability using classification and regression tool. Biophys Chem. 2007 Feb;125(2-3):462-70.

43. Huang MC, Seyer JM, Kang AH. Comparison and accuracy of methodologies employed for analysis of hydropathy, flexibility and secondary structure of proteins. J Immunol Methods. 1990 May 8;129(1):77-88

44. Inbar Y, Benyamini H, Nussinov R, Wolfson HJ (2003). Protein structure prediction via combinatorial assembly of sub-structural units. Bioinformatics. 19 Suppl 1:i158-i168.

45. Jacob E, Unger R. A tale of two tails: why are terminal residues of proteins exposed? Bioinformatics. 2007 Jan 15;23(2):225-230

46. Janin J. Surface area of Globular Proteins. J Mol Biol. 1976; 105(1):13-4.

47. Janin J. Surface and Inside Volumes in Globular Proteins, Nature. 1979;277(5696):491-2

48. Jiang F. Prediction of protein secondary structure with a reliability score estimated by local sequence clustering Protein Eng 2003; 16:651-657.

49. Jones DT. Protein secondary structure prediction based on position-specific scoring matrices. J Mol Biol 1999;292:195-202.

50. Jones DT, McGuffin LJ (2003). Assembling novel protein folds from super-secondary structural fragments. Proteins. 53 Suppl 6:480-485.

51. Kabsch W, Sander C. Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-bonded and Geometrical Features. Biopolymers 1983;22:2577-637.

52. Karplus K, Barrett C, Hughey R. Hidden markov models for detecting remote protein homologies. Bioinformatics 1998;14:846-856.

53. Kim H, Park H. Prediction of protein relative solvent accessibility with support vector machines and long-range interaction 3D local descriptor. Proteins 2004; 54:557–562.

54. Kitchen DB, Decornez H, Furr JR, Bajorath J. Docking and scoring in virtual screening for drug discovery: methods and applications. Nat Rev Drug Discov. 2004; 3(11):935-949.

55. Keller JM, Gray MR, Givens JA. A fuzzy K-Nearest Neighbor Algorithm. IEEE Trans on SMC 1985;15:580-585.

56. Kulkarni-Kale U, Bhosle S, Kolaskar AS. CEP: a conformational epitope prediction server. Nucleic Acids Res. 2005 Jul 1;33(Web Server issue):W168-71.

57. Lee B, Richards FM. The interpretation of protein structures: estimation of static accessibility. J Mol Biol. 1971; 55(3):379–400.

58. Lee J, Kim SY, Joo K, Kim I, Lee J (2004). Prediction of protein tertiary structure using PROFESY, a novel method based on fragment assembly and conformational space annealing. Proteins. 1;56(4):704-714.

59. Levenberg K. A Method for the Solution of Certain Problems in Least Squares. Quart. Appl. Math. 2, 164-168, 1944.

60. Levinthal C. Are there pathways for protein folding? Journal de Chimie Physique et de Physico-Chimie Biologique 1968 ; 65, 44.

61. Li X, Jacobson MP, Friesner RA (2004). High-resolution prediction of protein helix positions and orientations. Proteins. 55(2):368-82.

62. Li Z, Scheraga HA (1987). Monte carlo-minimization approach to the multiple-minima problem in protein folding. Proc Natl Acad Sci USA. 84:6611-6615.

63. Marquardt D. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. SIAM J. Appl. Math. 11, 431-441, 1963.

64. Mathews B. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochim Biophys Acta 1975;405:442-451.

65. Meiler J, Baker D. Coupled prediction of protein secondary and tertiary structure. Proc Natl Acad Sci 2003;100:12105-10.

66. Miller S, Janin J, Lesk AM, Chothia C. Interior and surface of monomeric proteins. J Mol Biol. 1987;196(3):641–656.

67. Moult J. A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction. Current opinion in structural biology. Curr. opin. struct. biol. 2005; 15(3): 285-289.

68. Moult J., Rigorous performance evaluation in protein structure modelling and implications for computational biology. Philos Trans R Soc Lond B Biol Sci. 2006 Mar 29;361(1467):453-8.

69. Naderi-Manesh H, Sadeghi M, Arab S, Movahedi AAM. Prediction of protein surface accessibility with information theory. Proteins 42: 452-459 (2001).

70. Novotny J, Handschumacher M, Haber E, Bruccoleri RE, Carlson WB, Fanning DW, Smith JA, Rose GD. Antigenic determinants in proteins coincide with surface regions accessible to large probes (antibody domains). Proc Natl Acad Sci U S A. 1986 Jan;83(2):226-30.

71. Pavlik P, Siegel RW, Marzari R, Sblattero D, Verzillo V, Collins C, Marks JD, Bradbury A. Predicting antigenic peptides suitable for the selection of phage antibodies. Hum Antibodies. 2003;12(4):99-112.

72. Pedersen JT, Moult J., Ab initio protein folding simulations with genetic algorithms: simulations on the complete sequence of small proteins. Proteins. 1997;Suppl 1:179-84.

73. Ring CS, Cohen FE (1993). Modeling protein structures: construction and their applications. FASEB J. 7(9):783-790.

74. Rost B. Review: protein secondary structure prediction continues to rise. J Struct Biol 2001;134:204-18.

75. Rost B, Sander C. Combining Evolutionary Information and Neural networks to Predict Protein Secondary Structure. Proteins: Structure, Function and Genetics 1994;19:55-72.

76. Rost B, Sander C. Conservation and prediction of solvent accessibility in protein families. Proteins 1994; 20:216–226.

77. Salamov AA, Solovyev VV. Prediction of Secondary Structure by Combining Nearest Neighbor Algorithms and Multiple Sequence Alignments. J Mol Biol 1995;247:11-15.

78. Salamov AA, Solovyev VV. Prediction of Secondary Structure using Local Alignments. J Mol Biol 1997;268:31-36.

79. Saraboji K, Gromiha MM, Ponnuswamy MN. Relative importance of secondary structure and solvent accessibility to the stability of protein mutants. A case study with amino acid properties and energetics on T4 and human lysozymes. Comput Biol Chem. 2005 Feb;29(1):25-35.

80. Sim J, Kim SY, Lee J. Prediction of protein solvent accessibility using fuzzy k-nearest neighbor method. Bioinformatics. 2005 Jun 15;21(12):2844-9.

81. Simons KT, Kooperberg C, Huang E, Baker D (1997). Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. J. Mol. Biol. 268(1):209-225.

82. Skolnick J, Kolinski A (1991). Dynamic Monte Carlo simulations of a new lattice model of globular protein folding, structure and dynamics. J Mol Biol. 221(2):499-531.

83. Sujatha MS, Balaji PV. Identification of common structural features of binding sites in galactose-specific proteins. Proteins. 2004 Apr 1;55(1):44-65

84. Thornton JM, Edwards MS, Taylor WR, Barlow DJ. Location of 'continuous' antigenic determinants in the protruding regions of proteins. EMBO J. 1986 Feb;5(2):409-13.

85. Venclovas C, Zemla A, Fidelis K, Moult J (2001). Comparison of performance in successive CASP experiments. Proteins. Suppl 5:163-170.

86. Venclovas C, Zemla A, Fidelis K, Moult J (2003). Assessment of progress over the CASP experiments. Proteins. 53 Suppl 6:585-595.

87. Wodak SJ, Janin J. Analytical approximation to the accessible-surface area of proteins . Proc Natl Acad Sci U S A. 1980; 77(4): 1736–1740.

88. Xu Z, Zhang C, Liu S, Zhou Y. QBES: predicting real values of solvent accessibility from sequences by efficient, constrained energy optimization. Proteins. 2006 Jun 1;63(4):961-6.

89. Yahyanejad M, Burge CB, Kardar M. Untangling influences of hydrophobicity on protein sequences and structures. Proteins. 2006 Mar 1;62(4):1101-6.

90. Yi TM, Lander ES. Protein Secondary Structure Prediction using nearest-neighbor Methods. J Mol Biol 1993;232:1117-1129.

91. Yu ZG, Anh VV, Lau KS, Zhou LQ. Clustering of protein structures using hydrophobic free energy and solvent accessibility of proteins. Phys Rev E Stat Nonlin Soft Matter Phys. 2006 Mar;73(3 Pt 1):031920

92. Yuan Z, Huang B. Prediction of protein accessible surface areas by support vector regression. Proteins. 2004 Nov 15; 57(3):558-64.

93. Zemla A, Venclovas C, Fidelis K, Rost B. 1999. A modified definition of SOV, a segment-based measure for protein secondary structure prediction assessment. Proteins. v34. 220-223.

94. Zhang X, Mesirov JP, Waltz DL. Secondary structure prediction with support vector machines. J Mol Biol 1992; 225:1049-63.

VITA

Rajkumar Bondugula is born March 31, 1980 in the city of Nizamabad, in Andhra Pradesh state of India. After his schooling at Vijay Public School, Nizamabad, Rajkumar went to Vignan Vidyalayam Junior College at Hyderabad. He received the Bachelor of Engineering degree in Computer Science and Engineering from University of Madras in May 2001. He then received Master of Science in Computer Science, with emphasis in Computational Intelligence and Vision, from University of Missouri-Columbia, Missouri, USA, in December 2003. He then received Doctor of Philosophy in Computer Science, with emphasis in Bioinformatics, in May 2007 from University of Missouri-Columbia. He is currently employed at Biotechnology HPC Software Applications Institute as a Research Scientist in Frederick, Maryland.