University of Alberta

Computational Prediction of Strand Residues from Protein Sequences
by

Kanaka Durga Kedarisetti

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Software Engineering and Intelligent Systems

Department of Electrical and Computer Engineering

Spring 2012
Edmonton, Alberta

This thesis is dedicated to my Husband

Abstract

Accurately identifying strand residues (β-residues) from protein sequences aids prediction and analysis of numerous structural and functional aspects of proteins. This thesis is focused on improving sequence-based prediction of strand residues and strands, which in turn would lead to better recognition of β-sheets (arrangements of multiple strands). We developed a novel ensemble-based predictor, BETArPRED, achieving a statistically significant performance improvement over existing, state-of-the-art secondary structure predictors. Our method improves prediction of strand residues and strands, and it also finds strands that were missed by the other methods. When compared with the top-performing three dimensional structure predictor, our BETArPRED improves predictions of strands and provides more correct predictions of strand residues, while the other predictor achieves higher rate of correct strand residue predictions when under-predicting strands. Next, we investigate strand residue-residue pair propensities incorporating long-range interactions, and a scoring function that uses these propensities. This scoring function is empirically shown to differentiate between strand and non-strand residues. We study the effect of residue conservation and directionality of strands in β-sheets on these propensities, and conclude that they provide little to no further improvement. We also compare our pair propensities with other recently proposed relative frequency-based pair propensities, and find that our pair propensities provide better discriminatory power in judging a residue from a strand to a non-strand. These proposed pair propensities could be used to further improve the sequence-based β-residue predictors.

Acknowledgements

First of all, I would like to offer my sincere gratitude to my supervisor Dr. Lukasz Kurgan for his constant guidance, encouragement and continuous support throughout my research. I am grateful to have Dr. Kurgan as my supervisor due to his passion for world-class research and his insight into the subject.

I also would like to offer my sincere thanks to my co-supervisor, Dr. Scott Dick, who has supported me throughout my research with his knowledge. Without his encouragement and effort, this thesis would not have been completed or written to this level of perfection.

I am sincerely thankful to the committees at UOA, NSERC, ICORE and all of all of those who supported me financially and in every other respect during the completion of the research. I also offer my sincere thanks to the BIOMINE research group, who contributed their assistance and discussions to the advancement of my research.

I am grateful for my family members mainly my loving husband Babu, lovely daughter Priyanka and kind son Pradyumna, who supported me in many ways which cannot be described in words and encouraged me to achieve my goal.

Lastly, my deepest gratitude goes to my father, for his care, love and encouragement throughout my education, putting me on a steady path of progress.

Thank you GOD for providing the above resources to accomplish my goal.

Table of Contents

List of Tables

List of Figures

Abbreviations

3D – Three Dimensional

AA – Amino Acid/Protein residue

Acc – Accuracy

ASSC – Average Strand Segment Coverage

BLAST – Basic Local Alignment Search Tool

BETArPRED – Beta residue Prediction

CASP – Critical Assessment of Protein Structure Prediction

CDHIT – Cluster Database at High Identity with Tolerance

CFS – Correlation Based Filter Method

CONS – Consistency Based Filter Method

DSSP – Dictionary of Secondary Structure of Proteins

FP – False Positives

FN – False Negatives

LIBLINEAR – Library for Large Linear Classification

LOG – Logistic Regression

ML - Machine Learning

NRBF – Normalized Gaussian Radial Basis function

MCC – Mathew's Correlation Coefficient

MSA – Multiple Sequence Alignment

PDB – Protein Data Bank

PSI-BLAST – Position Specific Iterative Basic Local Alignment Search Tool

PSSM – Position Specific Scoring Matrix

Residue_residue – interaction between residues (A_i_A_j)

SOV – Segment Overlap

SS- Secondary Structure

SVM – Support Vector Machines

TN – True Negatives

TP – True Positives

WEKA – Waikato Environment for Knowledge Analysis

1 Introduction

In molecular biology, a protein is a biologically active molecule, whose function *in vivo* is defined by its 3-dimensional (3D) structure. However, when it is first produced by a living organism, a protein is *not* in its 3D structure; it is instead a linear arrangement of amino acids, selected and ordered by the genetic code of that organism. This linear sequence must then undergo a complex process of folding to produce a stable 3D structure. If this complex process proceeds normally, the resulting biomolecule will correctly implement the target functionality. On the other hand, if the process goes awry, a wide spectrum of diseases can result.

In his Nobel Prize-winning work, Anfinsen (1973) experimentally showed that proteins form specific shapes determined by their amino acid sequence. Since then, sequence based computational structure prediction methods assume the protein sequence contains all of the information needed to predict the 3D structure. However, it is not yet known fully how the 3D structure can be determined from this sequence. Hence this topic has been a main focus of research in the last few decades. A full understanding of the relationship between protein sequence and structure would aid in the prediction of unknown protein structures and would impact related areas, such as rational design of novel proteins and peptides. Knowing the 3D structure is also essential to understanding a protein's function and to investigate the interactions with other molecules (Singh et al., 2006; Laskowski et al., 2005; Espadaler et al., 2005; Bowie et al., 1991), and is crucial in rational drug structure-based design (Klebe et al., 2000; Lengauer et al., 2000).

Structural bioinformatics, which is a subfield of computational biology, attempts to determine the (unknown) 3D structure of a new protein sequence, based on the

assumption that a given protein sequence folds into a unique protein structure (Anfinsen, 1973) and utilizing a principle that similar protein sequences lead to similar 3D structure. Currently, only a small fraction of all proteins have a known 3D structure. These structures are published and stored in the publicly available Protein Data Bank (PDB) (Berman et al., 2000). As of 15th January 2012, PDB includes 72,683 proteins (http://www.rcsb.org/pdb/results/), while the overall number of known non-redundant protein sequences that are stored in the RefSeq database (Pruitt et al., 2002) includes 14,090,554 (http://www.ncbi.nlm.nih.gov/RefSeq/). This wide structure-sequence gap is a result of the relatively low throughput of empirical methods for determining protein structure (e.g. X-ray crystallography, nuclear magnetic resonance).

The 3D (tertiary) structure of a protein consists of repeating units of secondary structure (SS) *states*, which include helix (h), strand (e) and coil (c) states. The two major types of secondary structures are the α-helix (helix) and β-strand (strand). About half of the amino acids (AAs) that comprise a protein sequence fold into the α-helix and β-strand secondary structures; the remaining residues are in more irregularly structured states called coils. The secondary structures are arranged in the 3D fold, which in turn defines the unique physical and chemical properties of proteins (Rost and Sander, 1996). Hence, determining the sequence of secondary structures in a protein is an important intermediate step in determining the tertiary structure. During the last three decades, there has been intense research in the sequence-based prediction of the protein secondary structures.

1.1 Motivation

In spite of significant advances in secondary structure prediction, the existing approaches for predicting strands from protein sequences are of relatively poor quality when compared to the other two structural types (α-helix and coil). One of the reasons for this is that strands can interact with other strands positioned far away in the linear protein sequence to form β-sheet structures. These long-range interactions are a unique feature of the strand structures, and are essential to understanding the structure of β-sheets (Zhang and Kim, 2000). Strand to strand interactions are wide-spread (every strand interacts with at least one other strand to form a β-sheet) and the linear distance between interacting strands is irregular. In contrast, helices and coils are established based on local interactions and they rarely interact with each other. Only a few methods have been proposed for the prediction of long-range interactions in a protein sequence and their accuracy is relatively low (Hubbard, 1994; Asogawa, 1997; Baldi et al., 2000; Steward and Thornton, 2002; Rost et al., 2003; Punta et al., 2005; Vullo et al., 2006; Cheng and Baldi, 2007), which implies that the prediction of strands is a difficult problem. Other reasons for the relatively low performance of the existing approaches are the weak coupling between β-residues pair's on neighboring strands in a β-sheet (Mandel-Gutfreund et al., 2001) and the lack of a systematic approach to solve this problem (Cheng and Baldi, 2005).

At the same time, an understanding of strand structures finds several important applications in the prediction of β-sheets and the prediction of the tertiary structure of proteins (Zaremba and Gregoret, 1999; Steward and Thornton, 2002; Ruczinski et al., 2002; Rost et al., 2003; Cheng and Baldi, 2005; Wu and Zhang, 2008; Lippi and Frasconi, 2009; Max et al., 2010), characterization of super-secondary structures and

protein folding patterns (Kamat and Lesk, 2007), elucidation of folding pathways and understanding the stability of protein folds (Smith and Regan, 1997; Merkel and Regan, 2000; Mandel-Gutfreund et al., 2001), design of new proteins (Smith and Regan, 1995; 1997; Kortemme et al., 1998; Kuhlman et al., 2003), and in investigations of certain mechanisms causing neurodegenerative diseases(such as Alzheimer's, Parkinson's and Prion diseases) (Fernandez-Escamilla et al., 2004; Stefani and Dobson, 2003; Stöhr et al., 2008; Kedarisetti et al., 2008). We also observe that more than 75% of proteins currently in the PDB contain β-sheets, demonstrating the importance of these structures. These applications and the abundance of the β-sheet structures motivate the need for computational approaches that improve the prediction of strand residues (amino acids in the protein sequence that fold into strands), strands (segments of consecutive β-residues), and their pairs (pairs of strands that form β-sheets) from the protein sequence.

1.2 Overview of proposed research

Before computational methods can identify β-sheet candidates, we must first identify the residues that make up individual strands; current computational methods are relatively ineffective for this task. Thus, our aim is to improve the prediction of strands from the protein sequence, and to investigate the pairings of the amino acids that are crucial for the formation of β-sheets. To this end, this thesis addresses the following three objectives:

Objective 1: Secondary structure predictors normally solve the three-state prediction problem. However, we can also cast secondary structure prediction as three two-state problems (strand vs. non-strand; helix vs. non-helix; coil vs. non-coil). This will be our approach in the construction of our strand residue prediction algorithm (BEATrPRED). In this research, we first compare two-state strand predictions against two-state helix

predictions from state-of-the-art secondary structure predictors. This is needed to verify the continued existence of the quality gap between strands and helices in this new representation.

Objective 2: We investigate the creation of a new, more accurate method for the prediction of the strand residues and strands. We hypothesize that improvements can be achieved by employing a consensus-based approach (by combining multiple existing prediction methods), and by combining local and long-range predicted structural information.

Objective 3: We investigate the propensities of the residue_residue pairs in the strand_strand contacts, and we hypothesize that these propensities can be used to further improve the prediction of the strand residues and strands. We also study the influence of the sequence conservation on these propensities.

In Objective 1, we compare a group of state-of-the art prediction models on a new dataset of 429 protein chains, which we created to minimize the effect of templates in the models (see Section 4.2.2 for further discussion). Protein chains in this dataset were selected from the PDB based on low similarity to each other, high-resolution structure determination, and recent deposition (at the time of the experiments). From this comparison, we select the best-performing predictors as the base predictors in our proposed ensemble in Objective 2. We combine predictions from these base methods with additional features (representing residue, window, and chain properties that are based on AA, SS and depth information) to form our proposed method. We then evaluate the performance of this new predictor against the individual predictors from Objective 1 on our new dataset. In addition, we compare all of these predictors, our new predictor,

and an additional method (the best-performing 3D structure predictor from the CASP 8 competition (Zhang, 2009)) on the CASP 8 competition dataset. In Objective 3, we empirically determine the propensity of amino acid pairs to align with each other in a strand pairing in a β-sheet. We computed these propensities based on the direction of alignment between the strands (parallel or anti-parallel). We also examine how conservation of amino acids interacts with these propensities. We then empirically determine whether these propensities can be used to identify strand residues.

1.3 Outline

The remainder of this dissertation consists of five chapters. Chapter 2 provides background information. Chapter 3 details our experimental design and evaluation procedures. Chapter 4 addresses objectives 1 and 2, summarizing and evaluating the model developed for the prediction of strand residues and strands. Chapter 5 addresses objective 3 and details the investigation of strand residue_residue pair propensities. Finally, Chapter 6 concludes with a summary of contributions and future work.

2 Background

In this chapter, we first provide background information on proteins that includes basic definitions of proteins, a brief overview of protein structures, related structural databases, details about β-sheet sub-units (candidates or topology), and the specific tools used to process protein sequence and structure data. We then describe the data mining concepts and methods that are used in the context of our research.

2.1 Background on proteins

2.1.1 Basic definitions

An **Amino Acid (AA)** is an organic acid containing an amino group ($^{+}NH_3$), carboxyl group (COO^{-}), variable side chain (R) group and a hydrogen atom (H) are all attached to a central α carbon (C_α) atom. The general structure of an amino acid is shown in Figure 2-1. There are 20 different types of amino acids that constitute the building blocks of proteins. Each amino acid has unique physiochemical properties that differ based on the variable side chain group. Amino Acids are denoted by a 3 letter code or a single letter code (Lodish et al., 2003). We employ the single-letter code in this dissertation. Amino acids also often referred to as residues.

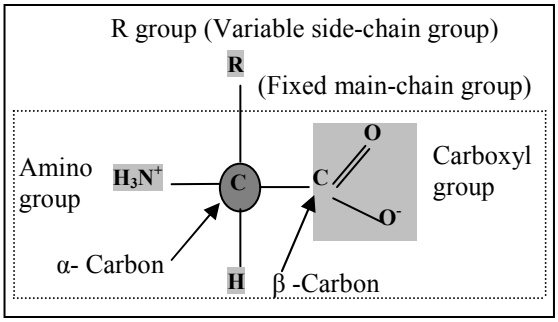


Figure 2-1 General structure of an amino acid

A **peptide bond** is a molecular bond between the carboxyl group of one amino acid and the amino group of the next amino acid with removal of a water molecule. This bond is also known as an amide bond (Petsko and Ringe, 2004). Figure 2-2 shows the peptide bond link between two amino acids.

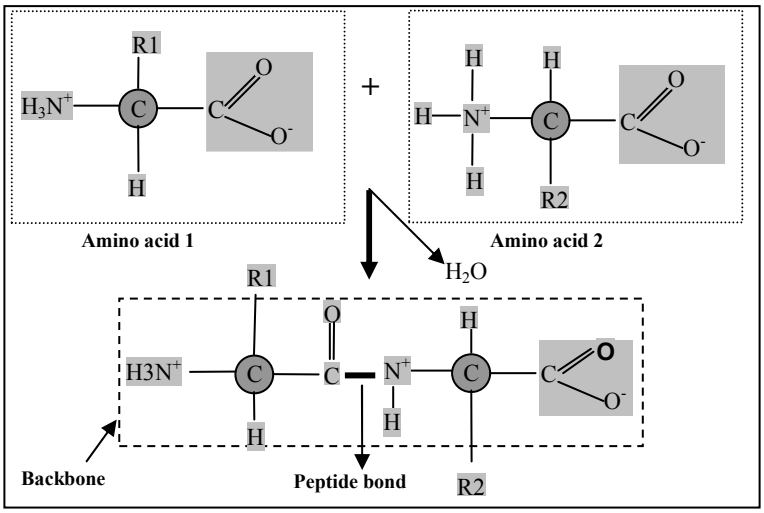


Figure 2-2 Peptide bond link between two amino acids

A **peptide** is formed by linking a series of amino acids via peptide bonds in a predefined order, usually <40 amino acids in length (Lodish et al., 2003). Figure 2-3 represents a peptide as a chain of amino acids, using a single letter code.

```
GIVEQCCTSICSLYQLENYCN
```

Figure 2-3 Peptide chain of human insulin (PDB id: 3Q6E, chain A)

A **polypeptide** is a chain of many (>40) amino acids linked by peptide bonds. Figure 2-4 shows a polypeptide chain representing amino acids using circles. The ends of the chain (called *terminals*) are annotated with the respective groups. The protein sequence is read from the N-terminal to the C-terminal (Lodish et al., 2003).

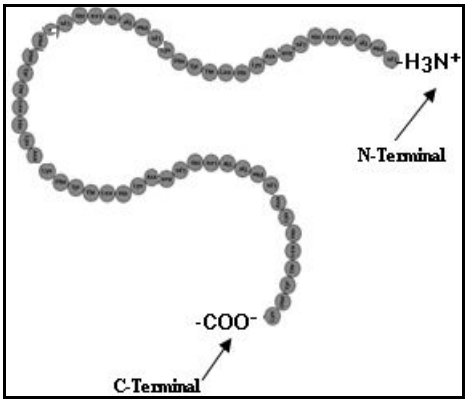


Figure 2-4 Polypeptide chain with the annotated terminals

The **N-terminal** refers to the end of a polypeptide chain terminated by an amino acid with a free amine group (NH3+) (Lodish et al., 2003). (see Figure 2-4).

The **C-terminal** refers to the end of a polypeptide chain terminated by an amino acid with a free carboxyl group (COO-) (Lodish et al., 2003). (see Figure 2-4).

Proteins are bio-molecules composed of one or more polypeptide chains containing amino acids linked together via peptide bonds (Petsko and Ringe, 2004).

The **Backbone** is a part of the peptide chain consisting of a series of N-Cα-Cβ atoms (main chain portion with sequence NCCNCCNCCNCC...), which helps to determine the protein conformation (shape) (Petsko and Ringe, 2004). See Figure 2-2.

Hydrogen bonds contribute significantly to the overall stability of the protein structure (its folded state). Hydrogen bonds are formed between main chain groups, or between the side chain groups, or between the main chain and side chain group; see Figure 2-1 for the definition of the main and side chain groups. In all cases, the hydrogen bond involves an attractive interaction between the hydrogen atom of a donor group (positively polarized), such as OH or NH, and a pair of nonbonding electrons on an acceptor group (negatively polarized), such as CO (Petsko and Ringe, 2004), see Figure 2-5. The donor group atom that carries the hydrogen must be electronegative for the attraction to be significant. The hydrogen bonds vary in length from 0.26 to 0.34nm (i.e., the distance between heavy atoms N and O in Figure 2-5).

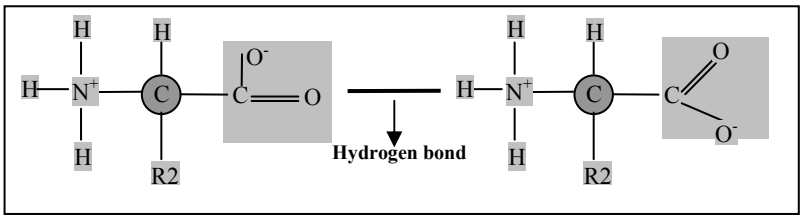


Figure 2-5 Hydrogen bond between main chain groups

Dihedral, Phi (ϕ) and Psi (ψ), angles. In Figure 2-6 the amino acid with the central carbon atom Cα forms a bond with N and Cβ atoms. The angle of the bond between Cα and N to the adjacent peptide bond is known as phi(ϕ) and the angle of the bond between Cα and Cβ to the adjacent peptide bond is known as psi(ψ) (Petsko and Ringe, 2004). Both of these angles, ϕ and ψ, form the dihedral angles/torsion angles (which describe conformations around rotatable bonds). These angles could take different values for the same amino acid occurring in different positions of the same protein sequence as well as in different proteins. They are useful to define the protein structure, particularly the secondary structure.

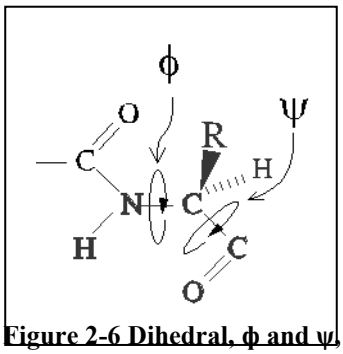


Figure 2-6 Dihedral, ϕ and ψ, angles

Multiple sequence alignment (MSA) arranges three or more sequences such that positions believed to be similar / identical based on functional, structural, or evolutionary relationships between these sequences are written in a common column. When a sequence does not possess a matching amino acid in a particular position then this position is denoted by a gap, i.e., "–". Multiple sequence alignment is useful for displaying the common fragments (segments of AAs) in a given set of sequences (Koonin and Galperin, 2003). These fragments may have common structure or function and thus one protein can be used to annotate other proteins.

```
VTISCTGSSSNIGAG-NHVKWYQQLPG
VTISCTGTSSNIGS--ITVNWYQQLPG
LRLSCSSSGFIFSS--YAMYWVRQAPG
LSLTCTVSGTSFDD-WYYSTWVRQPPG
```

[each letter represents one amino acid and '-' represents a gap]

Figure 2-7 Multiple sequence alignment example

Sequence profile or position-specific scoring matrix (PSSM) is a summary of the amino acid types present at each sequence position (column) in a given MSA; PSSM is also called a sequence profile. The matrix assigns positive scores to residues that appear more often than expected by chance and negative scores to residues that appear less often than expected by chance. It has been shown that for a given protein family (a set of functionally similar proteins), structural and functional constraints influence the amino acid types appearing at each position in the protein sequence and also that sequence–

structure correlations exist for secondary structures (MacCallum, 2004). Therefore, the sequence profiles are related to the secondary structure, i.e., they could be used and were used to predict the secondary structure.

Sequence homology

Homology is based on the evolutionary relationship found between two similar protein sequences which are derived from different species. Typically two protein sequences derived from the recent common ancestors are more similar than those derived from distant common ancestors since distant ancestors can accumulate many dissimilar mutations in their respective evolutionary paths. Hence, finding the evolutionary relationship between distant ancestors using only their protein sequences becomes a difficult task (Petsko and Ringe, 2004). Sequence homology is defined as the percentage of amino acids that are similar after aligning a protein sequence with other sequences. Proteins with similar function often (but not always) have similar protein sequences in corresponding functional regions. This observation has been used to find protein sequences with known structure/function that exhibit similarity to a protein sequence with unknown structure/function. Information about similar proteins provides insights into the structure and function of the query protein. This approach is also used to predict the protein secondary structures of newly discovered protein sequences, but requires that a query protein exhibits at least 30% homology to other proteins in the database. In the case of low similarity (i.e <25% similarity), predicting the unknown secondary structures becomes difficult. Tools like BLAST and PSI-BLAST have been developed to find the similar sequences from databases.

Sequence conservation

Scientists compare protein sequences from different species to identify residues that are conserved despite evolutionary change; it is assumed that such residues play particularly important roles in a protein's folding and function. However, sequences are only a 1D representation of 3D proteins. It is important to recognize that the 3D spatial orientation of residues also drives sequence conservation. For example, residue contacts in β-sheets, a binding surface, or an enzyme active site may have several conserved residues spread across the protein sequence, but in 3D space the residues are consolidated into a localized binding surface. In addition, amino acids can be divided into several groups of similar characteristics; substitution of an amino acid by a similar one can still result in a similar 3D structure (although this is not always the case) (Vyas et al., 2009; Cai et al.,2009).

2.1.2 Overview of protein structures

A protein is a large biomolecule consisting of a chain of amino acids (AA) that are linked through peptide bonds. This linear chain of amino acids can spontaneously fold into three dimensional structures called *native folds*, which are biologically active forms. The order of the AA's in a protein chain and the properties of their side chains determine the 3D-structure and function of the protein (Anfinsen, 1973; Rost and Sander, 1993). The structure of proteins can be analyzed on four different levels. These four levels are summarized below.

Primary structure (also called amino acid sequence or protein sequence) is defined by the linear order of AAs along a polypeptide chain in a protein (Branden and Tooze, 1999). The following Figure 2-8 depicts the primary structure of a protein using the single letter code for AAs.

```
MSLLNIKFTDNAVDYLKRREILDKILILITDDGGGKYSIQGGSCSMGAHFSIIWLDK
VDPDYPVKIANEQNVKIYTSDFDKTMLGPNMVMDYNAGSLSLSSDEGLLDGSVDIGN
GAALLKANKNVQMGINRQCEGHHHHHH
```

[The sequence continues over multiple rows]

Figure 2-8 Primary structure of Fe-S biosynthesis protein (PDB ID: 2QGO chain-A).

Secondary structure (SS) is defined by the localized sections of the folded segments of a polypeptide chain that are stabilized by regular patterns of hydrogen-bonds between the peptide NH and CO groups of different residues (Petsko and Ringe, 2004). The three basic types of secondary structures are the α-helix (h), β-strand (e), and coil (c), see Figure 2-9.

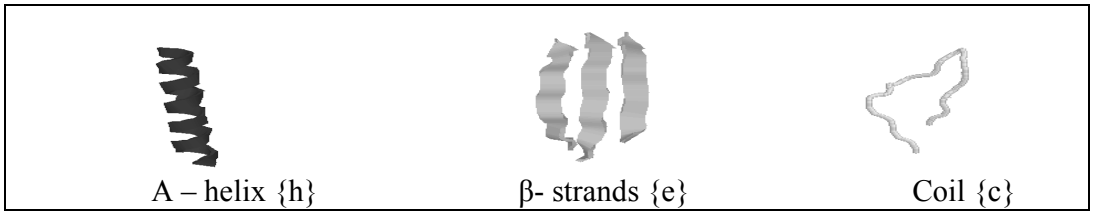


Figure 2-9 Secondary structures

These secondary structures form clusters (segments) in the primary structure, see Figure 2-10.

```
ccccceeeechhhhhhhhcccccceeeeeeccccccccccccccccceeeeeeecc
ccccccceecccccceeeechhhhcccccceeeeeeccceeeeeeccceeeeeeeeee
hhhhhhhhhhcccccccccccccccc
```

[h-helix state, e-strand state, and c-coil state]

Figure 2-10 Secondary structure of Fe-S biosynthesis protein (PDB ID: 2QGO chain-A).

An **α-helix** is formed when a polypeptide chain arranges into a regular spiral or rod like structure. In the most frequent type of helix structure, i.e., α-helix, the CO group of the backbone of an amino acid at position *j* forms a hydrogen bond with the NH backbone group of the amino acid that lies at position $j + 4$ (Petsko and Ringe, 2004). Hence, α-helix has four residues per turn. Other types of helical structures such as 3_{10} helix and π

helix, which are relatively rare in proteins, have three and five residues per turn, respectively.

A **β-strand** is usually three or more residues in length. Backbone atoms in two strands connect through hydrogen bonds and form a sheet. A stretch of at least three consecutive strand residues is identified as a β-strand if all the (φ, ψ) values in this region lie within the region defined by: $180° < \varphi < -30°$, $60° < \psi < 180°$ or $-180° < \psi < -150°$ (Gunasekaran et al., 1998). The other type of beta structure is β-bridge, defined by two backbone hydrogen bonds that will be described in more detail later.

A **Coil** is a non-repetitive, relatively irregular secondary structure. Three main types of coils include turns (which assume a few defined structures), bends and loops that are irregularly shaped (Branden and Tooze, 1999). Coils connect α-helix and β-strand segments, i.e., they serve as linkers, and without them proteins would be loosely packed.

Super secondary structures (also called motifs) involve multiple secondary structures in a particular geometric arrangement. If a single secondary structure is considered as a 'unit' then a super secondary structure would be comprised of at least two 'units' of secondary structure (Gruber A, et al., 2008). Some of these super secondary structures are known to have a specific biological role but for others their role is unknown.

The **tertiary structure** of a protein is a specific three-dimensional shape resulting from the folding of the entire polypeptide chain and can be seen as the spatial arrangement of the secondary structures (Branden and Tooze, 1999). The tertiary structure is defined by the coordinates of the atoms of the constituent AAs. The tertiary structure of the Fe-S

biosynthesis protein (PDB ID: 2QGO) is shown in Figure 2-11. This structure is shown in a simplified ribbon form that shows secondary structures of the protein backbone instead of the atomic coordinates.

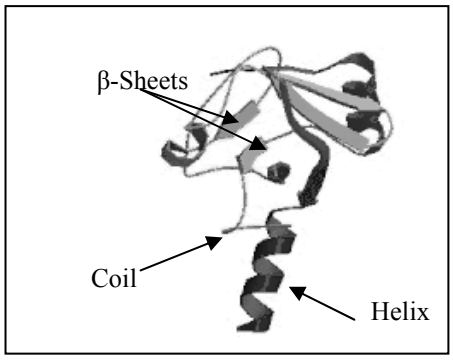
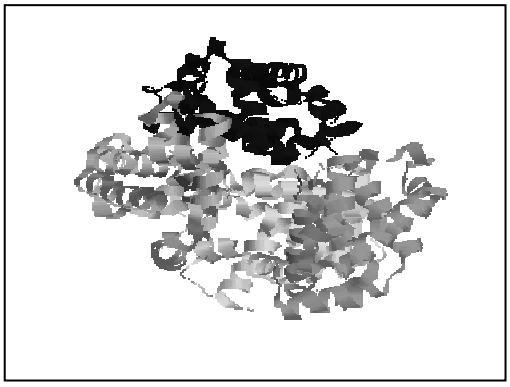


Figure 2-11 Tertiary structure of Fe-S biosynthesis protein (PDB ID:2QGO)

Quaternary structure is the arrangement of multiple folded protein chains in a multi-subunit complex (Branden and Tooze, 1999). A variety of bond interactions including hydrogen bonding, salt bridges, and disulfide bonds hold the various chains in a particular geometry. For example, the simplified ribbon representation for the quaternary structure of the deoxy hemoglobin (PDB ID: 1O1J) shown in Figure 2-12 contains four protein chains that together form a globular protein (i.e., protein that assumes a sphere-like shape).



[Each of the four constituent chains is shown using a different shade of gray]

Figure 2-12 Quaternary structure of Deoxy hemoglobin (PDB ID: 1O1J)

2.1.3 Protein databases

Protein Data Bank

The Protein Data Bank (PDB) is a worldwide repository of 3D structural information concerning proteins, which has identified structures for around 72,000 protein structures among the millions of known sequences. These structures are obtained by the experimental methods such as X-ray crystallography or NMR spectroscopy. The data is submitted by biologists and scientists from around the world, and is curated by the Research Collaborators for Structural Bioinformatics (RCSB) PDB Advisory Committee. The PDB repository can be accessed online for free (http://www.rcsb.org/). This database is well maintained and up-to-date, and has been used in many structure-related investigation applications. As of 15th January 2012, PDB contains 72683 protein chain structures (http://www.rcsb.org/pdb/results/). Several databases have been are derived from PDB to classify proteins in terms of their structure, function and evolution. PDB is a key resource in structural biology studies and is the source of data that is used to design and validate our proposed method.

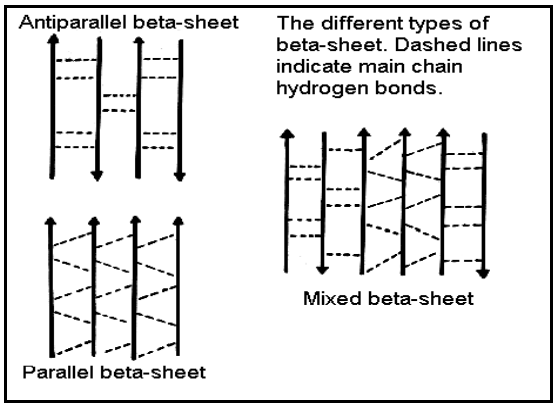
Dictionary of Secondary Structures of Proteins (DSSP)

The secondary structure label for each AA is assigned using the dictionary of secondary structures of proteins (DSSP) program (Kabsch and Sander, 1983), which uses the atomic coordinates of a given protein chain structure obtained from the PDB. The current version of this product is 17.1. The DSSP annotates each residue as belonging to one of the eight secondary structure types: H (α-helix), G (3-helix, also known as 3_{10} helix), I (5-helix, a.k.a. π-helix), B (residue in isolated beta-bridge), E (extended β-strand), T (hydrogen bond turn), S (bend), and "_" (any other). Typically, these eight states are reduced to

three states as follows: helix (h, which includes H, G, and I), strand (e, which includes E and B), and coil (c, which includes remaining types) (Moult et al., 2009). These three states are widely used to indicate the protein secondary structure states and are also used by the EVA (evaluate the accuracy of automated protein secondary structure prediction methods) web server (Rost and Eyrich, 2001).

2.1.4 β-sheets

A **β-sheet** is an assembly of two or more β-strands that are hydrogen bonded to form a sheet-like (planar) arrangement. The formations of backbone hydrogen bonds between adjacent strands which may be far away from each other in the sequence (Branden and Tooze, 1999) provide a significant increase in overall stability of a protein. Formation of β strand pairs can be subdivided into individual interactions between β-strand residues (known as β-residue pairs/ β-contacts), which allows for step-wise prediction of β-sheets (Cheng and Baldi, 2005). In β-sheets, the strand pairs can be arranged in three ways: antiparallel, parallel, and mixed, as shown in Figure 2-13 and discussed below.



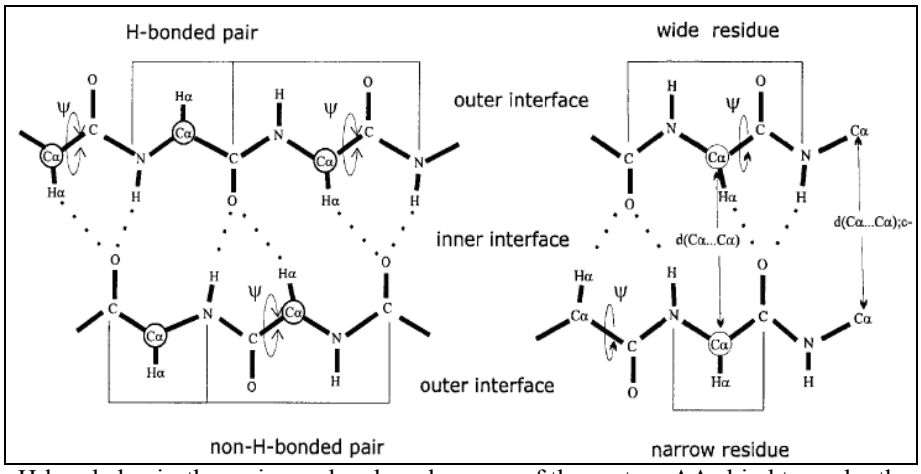
[Arrows denote β strands and their orientation in the protein chain and dashed lines denote hydrogen bonds]

Figure 2-13 Parallel, antiparallel. and mixed β-sheet arrangements.

Antiparallel β-sheets: In an antiparallel arrangement, the successive β strands run in opposite directions in the sequence so that the CO and NH groups of one residue in the first strand establish their hydrogen bonds with the NH and CO groups of the same partner in the second strand, respectively (Berg et al., 2002). This is known as a close pair of hydrogen bonds (Ho, 2002). Hence, in antiparallel β-strand pair data, either both residues of a pair are hydrogen bonded or both residues are non-hydrogen bonded (see Figure 2-14). This arrangement produces the strongest inter-strand stability as it allows for the inter-strand hydrogen bonds between carbonyls and amines to be planar, which is the preferred orientation. Hence, in antiparallel strand arrangement, N-terminus of one strand is adjacent to the C-terminus of the other strand.

Parallel β-sheets: In a parallel arrangement, the successive β strands run in the same direction. The NH group of one residue in the first strand is bonded to the CO group of a partner residue in the second strand. However, the CO group of the residue in the first strand is not bound to the NH group of the partner residue in the second strand, but instead finds a residue immediately following the other residue of the pair in the second strand (Berg et al., 2002). This is known as a wide pair of hydrogen bonds (Ho, 2002) as shown in Figure 2-14. This orientation is slightly less stable because it introduces non planarity in the inter-strand hydrogen bonding pattern. Hence, in parallel β-strand pair data, one residue is hydrogen bonded whilst the pairing residue is non-hydrogen bonded (see Figure 2-14).

Mixed β-sheets: In a mixed arrangement, an individual strand may exhibit a mixed bonding pattern, with a parallel strand on one side and an antiparallel strand on the other (Berg et al., 2002). Such arrangements are less common than a random distribution of orientations would suggest.



[In an H-bonded pair, the amine and carboxyl groups of the partner AAs bind to each other. In a non-H-bonded pair, such bindings do not exist; the amine and carboxyl groups bind to partner AAs in a *different* β-strand]

Figure 2-14 Different forms of bonding between partner AAs in a strand pair. A residue pair may be H-bonded or non-H-bonded (depicted on the left side of the figure). Additionally, H-bonded pairs may form with wide or narrow residues, depending on the orientation of the two strands (depicted on the right side of the figure) (Ho, 2002).

β-bridge

Formation of a single pair β-sheet hydrogen bond is defined as a β-bridge or β-bulge, i.e. a β-strand pair of length 1. DSSP defines bridge partners as residues across from each other on adjacent β-strands, and it also determines whether the bridge partners interact via backbone N–OH-bonds (in other words, a β-bridge must be an H-bonded pair). In general, a β-sheet consists of parallel or antiparallel bridges (Kabsch and Sander, 1983). Table 2.1 shows an example of a protein sequence, with the β-sheet subunits labeled.

Table 2-1 β-sheet subunits of the Fe-S biosynthesis protein (PDB ID:2QGO_A)

[where B denotes a β-bridge, p_i denotes parallel β--strand pairs, a_i correspond to antiparallel β--strand pairs, and i denotes strand pairs. The sequence continues over three rows in the table.]

Primary sequence **8-State Secondary structure** **β-Bridge** **Parallel β-strands pair** **Antiparallel β-strands pair**	MSLLNIKFTD NAVDYLKRRE ILDKILILIT DDGGGKYSIQ GGSCSMGAHF CCCCBEEECH HHHHHHHHTT CTTSEEEEEE CSSCSTTCCC CCCCCCCCCE **B** EEE EEE **p_1** **p_2** EEEEEE E **a_1**
Primary sequence **8-State Secondary structure** **β-Bridge** **Parallel β-strands pair** **Antiparallel β-strands pair**	SIIWLDKVDP DYPVKIANEQ NVKIYTSDFD KTMLGPNMVM DYNAGSLSLS EEEEESSCCT TCCEECBCSS CCEEEECHHH HTTSCSSEEE EEETTEEEEE **B** EEEE EEE EEE **p_3** **p_2** **p_1** EEEEE EE EE EEE EEE EE**EEE** **a_1** **a_2** **a_2** **a_3** **a_3** **a_4**
Primary sequence **8-State Secondary structure** **β-BRIDGES** **Parallel β-strands pair** **Antiparallel β-strands pair**	SDEGLLDGSV DIGNGAALLK ANKNVQMGIN RQCEGHHHHH H ETTEEEEEEE EEEEHHHHHH HHHHHCCCCC CCCCCCCCCC C EEEE **p_3** **E** EEEE **a_4** **a_4**

2.1.5 Tools used for generating datasets and features

CD-HIT:

CD-HIT(Li and Godzik, 2002) stands for Cluster Database at High Identity with Tolerance. CD-HIT takes a formatted sequence database as input and reduces the overall size of the database by removing 'redundant' (highly similar) sequences and outputs a set of 'non-redundant' (at a given similarity level) representative sequences. CD-HIT clusters all input sequences into groups with sequences that are similar with each other above a certainty identity threshold and then selects one chain from each cluster to generate the set of representative sequences. The algorithm implements a very fast heuristic to find highly similar segments between sequences to avoid costly full alignments.

BLAST AND PSI-BLAST

BLAST (Basic Local Alignment Search Tool) is a search method (Altschul, 1990) that finds sequences in a database similar to a given query protein sequence, where the

similarity exceeds a preset threshold score. Position-Specific Iterative BLAST (PSI-BLAST) is an iterative alignment method that uses sequence profiles (Altschul et al., 1997). The first iteration of the PSI-BLAST is similar to a run of the BLAST program and generates a MSA using the BLAST program output to calculate a PSSM. This PSSM is used by the second iteration of the PSI-BLAST to detect sequences in a database that are also above the threshold score. These newly found sequences are used to recalculate the PSSM. This process is repeated until no more new similar sequences are found or a user-defined number of iterations have elapsed. Whereas BLAST generates sequences that are similar based on a single query protein sequence, PSI-BLAST is able to retrieve sequences that have similar structure/function to the input sequence through a profile search that combines the underlying conservation information. Hence, PSI-BLAST method can identify related sequences even though much of their primary sequences have been altered through evolutionary changes (Altschul et al., 1997; Aravind and Koonin, 1999). PSI-BLAST tool is available online from the National Center for Biotechnology Information (NCBI) website (http://www.ncbi.nlm.nih.gov/BLAST/).

RDpred

Residue depth quantifies how deeply a given residue is buried within the protein 3D structure. This information aids in the prediction of protein folds, functional sites and in protein design. RDpred is a recent sequence based residue depth prediction method (Zhang et al., 2008) that predicts residue depths using three depth indices/definitions: two distance-based depths based on the MSMS (Koh et al., 2003) and DPX (Pintar et al., 2003) methods, and a volume based depth based on the SADIC algorithm (Varrazzo et al., 2005). These three approaches are complementary to each other. Since, the absolute correlations between these depth predictions range between 0.63 and 0.77 (Yuan and

Wang, 2008). In addition, RDPred was shown to outperform a competing sequence based residue depth predictor designed by Yuan (2008). The prediction of exposed residues (residues with low depth) has implications in characterization/prediction of interactions with ligands and other proteins, while the prediction of buried residues (residues with high values of depth) could be used in the context of the prediction and simulation of protein folding.

2.2 Background on computational methods

Experimental techniques for determining a protein structure such as X-ray Crystallography and Nuclear Magnetic Resonance (NMR) methods (Rhodes, 2006) remain slow, laborious, expensive and do not scale up to current sequencing speeds. Furthermore, using experiments to determine how proteins function is a daunting task; and furthermore their native environments are very specific, which can be difficult to replicate in the laboratory. Hence researchers have developed several high throughput computer methods that can rapidly sift through massive amounts of data and help determine the structure and function *in silico*. Machine learning methods are one of the computational approaches that aim to extract information from data through a process of training from examples. These approaches rely heavily on similarity of protein sequences for prediction of protein 3D structure (structural features, topology and coordinates). Machine learning methods are suitable candidates due to the abundance of data and a lack of a clear theoretical model that can be used to deduce structure.

2.2.1 Definitions

A **feature** describes an attribute or property of an object (i.e. it is an observation or computed value). The set of possible values that a feature can take is its domain. An object can be described by a set of features.

Example-2.1:

PDB_ID	2JWU
Chain_ID	A
Amino Acid Sequence	TTYKLILNLKQAKEEAIKELVDAATAEKYFKLYANAKTVEGVWTYKDETKTFTVTE
AA Sequence length	56
'T' count in AA sequence	9
'H' count in AA sequence	0

In Example 2.1, chain ID, length of the AA sequence, count of AAs of type T, and count of AAs of type H are the features that describe a given protein chain.

These feature values are summarized as, Chain_Id = A is a nominal feature, i.e., the values of this feature, which is "A", is nominal. Sequence length = 56 is a numerical feature, i.e., the values of this feature, which is "56", is numerical. Also, Amino acid 'T' count= 9 and Amino acid 'H' count= 0 are numerical features. Similarly, we can derive different types of features from the protein sequences.

A **dataset** is collection of objects, where each object is described by the same set of attributes/features. The most popular representation of a dataset is a two-dimensional table. A given row represents an object, and a column represents a feature.

Example-2.2: For the following three protein sequences, Table 2-2 represents a dataset, where each row represents an object/instance and a column is described by a feature.

sequence_1: TTLYTSLHAYFVAAPTGCNAEGFFATLGGEI
sequence_2: GCLGDKCDYNNGCCSGYVCSRTWKWCVLNGPW
sequence_3: MGINTRELFLNFTIVLITVILMWLLVRSYQY

Table 2-2 Protein dataset using feature-based representation.

Protein_length	Number of 'T's in AA sequence	Number of 'L's in sequence	
31	5	3	<- feature vector for sequence_1
32	1	2	<- feature vector for sequence_2
31	3	5	<- feature vector for3 sequence_3

Feature space is the cross-product of all the feature domains; this is the universal set from which examples are drawn.

Example-2.3: For the following protein sequence, protein is specified by a set of three features. The cross product of these features is the feature space.

sequence - TTLYTSLHGYFVFGPTGCNLEGFFATLGGEI

feature space - (Protein_length, # of 'T's in sequence, # 'L's in sequence)

Feature vector represents an object as one point in the feature space described by the values of its features. In machine learning feature vector is also called tuple/record/example, and is usually represented by one row in the dataset table.

Example-2.4: For the following protein sequence, generated three features that represent the protein instance, values of its features forms a feature vector,

Sequence - TTLYTSLHGYFVFGPTGCNLEGFFATLGGEI

Feature space- (Protein_length, # of 'T's in sequence, # 'L's in sequence)

Feature vector - (31,5,4)

Class (*C*) (also called predicted feature) is defined by the user and usually is a distinct feature. The values of the class feature are referred to as class labels. The labels reflect

some categorization of the underlying objects. The learning task in *classification* is to determine a mapping from feature vectors to class labels that approximates the observed label assignment with minimal error.

Training data is a finite subset of the entire available dataset defined as T_r=(A, C), which consists a subset of instances A =(A_1 . . . A_n) associated with class labels C =(C_1 . . . C_n), in which class is predefined. Training dataset is used to generate a prediction model using a learning process.

Test data is a finite subset of the entire available dataset defined as T_e= (A, C), which consists a subset of instances A =(A_1 . . . A_n) associated with class labels C =(C_1 . . . C_n), in which class is hidden from the prediction model. Test data that is used in a classification (prediction) process as well as in the learning process are known as in-sample evaluation. Test data that is used in a classification (prediction) process, and is not used in the learning process are known as out-of-sample evaluation. The prediction model uses the predicting features from the test dataset to predict the class labels, which are compared with the hidden (true) values to evaluate the predictive quality of the model. Out-of-sample evaluation is considered a less-biased estimate of the expected performance of the classifier (predictor) on all possible inputs from the feature space.

Example-2.5: A protein dataset shown in Table 2-3 includes class feature called protein_type (this feature will be predicted using the remaining features) which has two class values (labels) defined as peptide (P) or polypeptide (O). The protein length, number of 'T' in the AA sequence, and number of 'H' in the secondary structure are the predicting features. The training data is the first four rows, where the class labels are predefined. The last row where class is hidden and has to be predicted is the test data.

Table 2-3 Protein dataset using feature-based representation including class

	Predicting features			Class feature	
	Protein_length	**Number of 'T's in AA sequence**	**Number of 'H's in secondary sequence**	**Protein type**	
Sequence_1	31	5	5	P	Training dataset
Sequence_2	31	7	6	P	
Sequence_3	62	3	21	O	
Sequence_4	77	9	4	O	
Sequence_5	65	1	4	--	Test dataset

Positive and negative examples are the examples in a dataset that satisfy the class condition are called positive (+ve) examples and remaining are called negative (-ve) examples.

Example-2.6: The positive and negative examples are shown in Table 2-4, where class label 'P' is defined by a condition that needs three predicting features: protein_length, number of 'T's in AA sequence, and number of 'H's in secondary sequence (i.e. Length<40 and T>4 and H>4) Positive examples are the first two rows that satisfy the condition and negative examples are last two rows that do not satisfy the condition.

Table 2-4 Protein dataset using feature-based representation including class with type annotation.

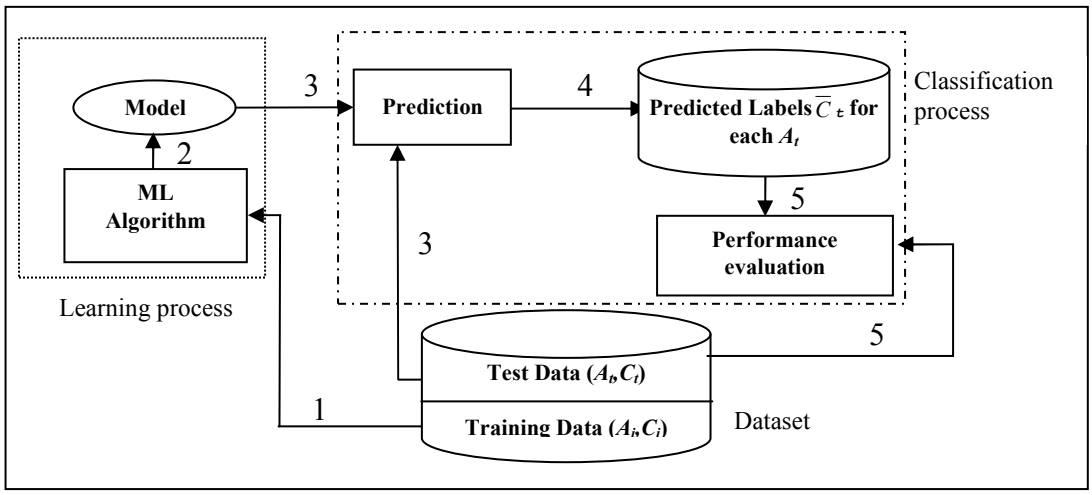
[Positive examples correspond to peptides (P) and negative examples correspond to others]

	Protein_length	**Number of 'T's in AA sequence**	**Number of 'H's in secondary sequence**	**Protein type**	
Sequence_1	31	5	5	P	+ve examples
Sequence_2	31	7	6	P	
Sequence_3	62	3	21	O	-ve examples
Sequence_4	77	9	4	O	

2.2.2 Classification methods

Machine learning (ML) in computational biology focuses extensively on the prediction of protein secondary structure or other unknown properties, based on *known* properties (Baldi and Brunak, 2001). This section is focused on ML methods that are used in this dissertation for the prediction of protein secondary structure. These methods concern the prediction/classification problem.

Classification is a task in which a classification system learns patterns from training dataset (T_r) and generates classification model that is used to predict a class label C_t for each object A_t from a test dataset (T_e). The classification model's performance is evaluated by comparing the predicted class labels with the original class labels for the instances in T_e using metrics and testing procedures. The entire process is summarized Figure 2-15.



[Numbers denote the order of operations.]

Figure 2-15 Learning and classification process, where training dataset used for learning and test data is used for classification (single-split with out of sample approach).

We explore the use of three classifiers in this dissertation: logistic regression (LOG) (Cessie and Houwelingen, 1992), a normalized Gaussian radial basis function (RBF) network (Bugmann, 1998), and a linear-kernel based Support Vector Machine (SVM) (Fan et al., 2008), which are described next. Our ultimate solution is based on the logistic regression; the other two classifiers were empirically evaluated to be inferior for our classification task. Thus, we provide a detailed introduction with an example for the LOG model, while the other two models are overviewed briefly. A more detail description of these classification methods is outside of the scope of this dissertation.

Logistic Regression (LOG)

Logistic regression (Cessie and Houwelingen, 1992) has been applied to a broad range of problems in computational biology such as protein interaction prediction (Qi et al., 2006), sequence-based prediction of DNA-binding residues (Hwang et al., 2007), prediction of protein intrinsic disorder (Peng et al., 2006), protein structural class prediction (Kedarisetti et al., 2006), etc. This record indicates that LOG is a reasonable candidate for our problem. LOG is useful when a user wants to predict the presence or absence of a characteristic or outcome based on values of a set of predicting variables. LOG fits a linear combination of predictor variables that passes through a sigmoid function, where the values of the class variable are the outcome. In our work, we used binary LOG where the class feature has two categories. A general form of the binary regression model is given in equation [2.1]:

$$P = \frac{1}{1 + e^{-z}}, \text{ where } z = C_0 + C_1 x_1 + C_2 x_2 + \ldots + C_m x_m \qquad [2.1]$$

Where, C_0 is a constant, C_i are coefficients of the linear polynomial z in the predicting features x_i, $i = 1, 2, \ldots, m$ is the feature index, and m is the number of predicting features.

The value of z is used to compute the probability P for a given outcome (label). The probability of the second outcome (in the binary classification) equals 1– P. The coefficients C_0 and C_i are calculated from the training dataset to minimize error rate on that dataset. We use WEKA (Hall et al., 2009) to derive the coefficients; a more detailed explanation of this calculation is outside of the scope of this dissertation.

The logistic regression is demonstrated below with an example using the data shown in Table 2-3. This simple model uses three predicting features to predict the two class labels defined by the protein type class features. The LOG model is generated by fitting the training data (the first four rows). The resulting coefficients are summarized in Table 2-5.

Table 2-5 LOG coefficients from the training dataset from Table 2-3

Variables	Coefficients	Coefficient values
Protein length (x_1)	C_1	-0.7034
Number of 'T's in AA sequence (x_2)	C_2	-0.5974
Number of 'H's in protein Secondary sequence (x_3)	C_3	-0.7886
	C_0	46.0905

We use this model to predict the last (test) instance sequence_5 in Table 2-3. The probabilities P of each class label are calculated using the equation [2.1] as follows

z = (-0.7034*65) + (-0.5974*1) + (-0.7886*4) + 46.0905 = -3.3823

$$P_{\text{label P}} = \frac{1}{1+e^{-z}} = 0.0329$$

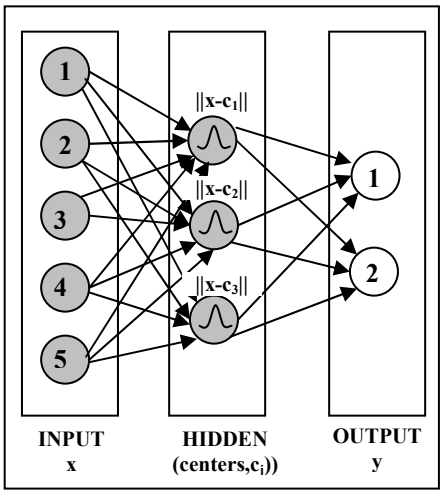
$$P_{\text{label O}} = \frac{1}{1+e^{z}} = 0.9672$$

$P_{\text{label P}} + P_{\text{label O}}$ = 0.0329+0.9672 = 1

Hence, this test instance is classified as class O with 0.9672 probability and error of 0.0329.

Normalized Gaussian radial basis function network (NRBF)

Neural Networks (NNs) are widely used in protein secondary structure prediction problem and also the earliest machine learning technique applied in the field of computational biology (Stormo et al., 1982). An example of an artificial neural network containing radial basis functions is shown in Figure 2-16, where there are three layers in the network, the input layer, the output layer, and a hidden layer. Each hidden node has a different radial basis function that is centered on a feature vector from the training dataset. The goal of this type of network is to create a model that correctly maps the inputs to the output using training data, so that the model can be used to produce the output class label when it is unknown (Abdi, 1994). In normalized RBF network (Bugmann, 1998) the output node activity is normalized by the total input activity in the hidden layer. As a result of the normalization, the activity of the hidden nodes determines which weights contribute the most to the output.

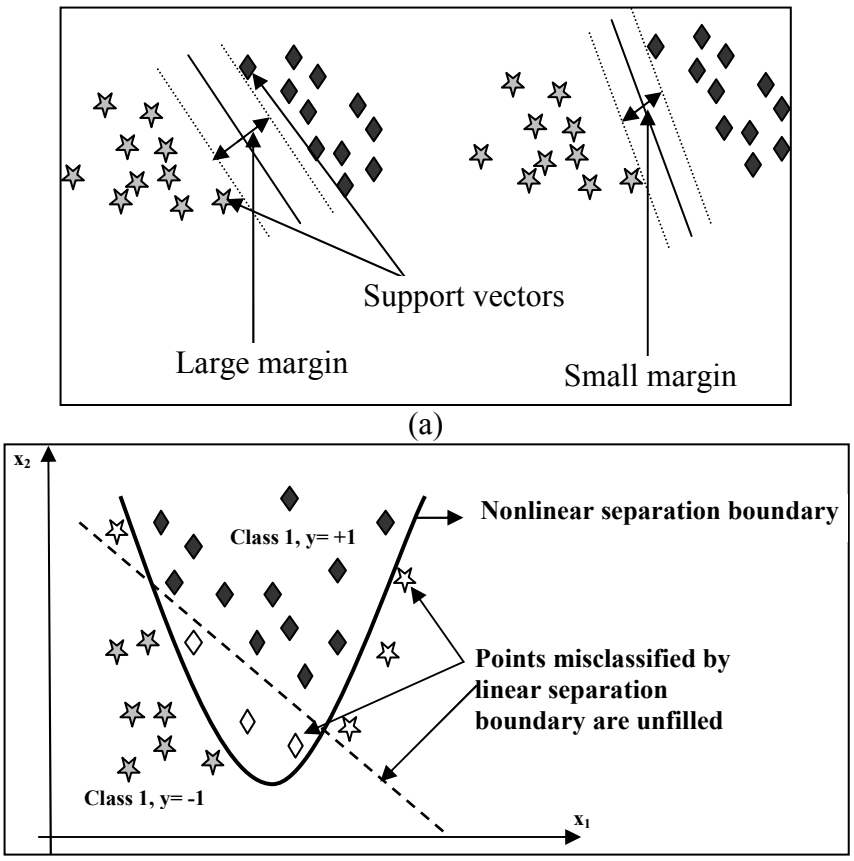


[This diagram is redrawn from the paper (Abdi, 1994)]

Figure 2-16 Radial basis function NN model

Support Vector Machines (SVM)

The goal of SVM modeling is to find the optimal hyper plane that separates feature vectors into classes. Figure 2-17(a) shows, a simple SVM model in a binary class, where the feature vector classes are represented by diamonds and stars. SVM model detects a single line from out of all possible lines that separates the feature vectors into two classes. The feature vectors near the separating line are called support vectors and the distance between the dashed lines that are drawn parallel to the separating line and close to the support vectors is called the margin. SVM detects the single optimal line by maximizing the width of the margin. If the data is not separable by linear plane, SVM with the help of kernel mapping function transform the data into a higher dimensional space in order to perform the separation. In Figure 2-17(b) shows, SVM using non-linear polynomial kernel function to transform the data into a higher dimensional space to perform the separation (http://www.dtreg.com/svm.htm). The most common kernels for a range of applications are linear, polynomial, radial basis function (RBF), and sigmoidal. In this work we use SVMs with linear kernels to perform the prediction of binary class feature vectors. More details about SVM can be found in numerous sources (Cristianini and Shawe-Taylor, 2000; Smola et al., 2000; Campbell, 2002; Sanchez, 2003; Fan et al., 2008).



[source of this diagram is from http://www.dtreg.com/svm.htm and redrawn]

Figure 2-17 (a) shows separating the points with straight line and (b) shows separating the points with non-linear curve (polynomial)

2.2.3 Feature selection methods

Feature selection methods are used to reduce dimensionality of data, i.e., the number of the predicting features. A feature selection method simplifies the subsequently used classifier by retaining only the relevant features. This may lead to improving the accuracy of the classifier and it decreases the size of the dataset. In this dissertation, we consider two feature selection strategies: a filter-based and a wrapper based method.

Filter based methods remove unnecessary features without using a classifier (Kohavi and John, 1996). Some filter based methods strive for retaining consistency in the data

(Allmuallim and Deitterich, 1991), i.e., they remove a feature only when doing so would not worsen the consistency. Consistent data are when a given combination of features and their values is associated with a single class label. Instances are considered inconsistent if they have the same feature values and different class labels. Other filter based methods rank features according to a relevancy score (Kira and Rendell, 1992; Holmes and Nevill-Manning, 1995). In our work, we considered two filter-based methods that are computationally efficient, which is important given the relatively large size of our data. The consistency-based (CONS) method (Liu and Setiono, 1996) is monotonic, fast, able to remove redundant and/or irrelevant features, and capable of handling some noise. The correlation-based (CFS) method (Hall, 2000) is a fast filter method that identifies relevant features as well as redundancy among relevant features within the high dimensionality data. These two methods were shown to reduce the dimensionality of the feature vector while maintaining or improving prediction quality in the subsequent classification (Liu and Setiono, 1996; Hall, 2000).

The **CONS method** (Liu and Setiono, 1996) uses a ratio between the numbers of inconsistent vs. total number of examples when the input data are projected onto a given subset of features. The CONS method attempts to amplify the discriminating power of the data, as defined by the predicting features. This method finds the smallest set of features that can distinguish classes, as if with the full feature set.

The **CFS method** (Hall, 2000) uses correlation as the relevancy score. This score considers the correlation between a given feature and the class feature and the correlation of that feature with other predicting features. The CFS method uses a ratio between a correlation-based estimate of the predictive value of a given feature (correlation with the class) set and its estimated redundancy (with other predicting features). For example, this

correlation can be calculated as the entropy of Y before and after observing X. The relevance score measure computed by CFS value lies between 0 and 1. A value of 0 indicates that X and Y have no association; the value 1 indicates that knowledge of X completely predicts Y. The CFS method uses a search algorithm along with it's correlation-based relevancy score to evaluate the merit of feature subsets.

The **wrapper-based method** (Hall and Smith, 1999) uses a classification algorithm in a cross validation (CV) design to evaluate feature subsets. In other words, feature sets are assessed based on their prediction quality using a given classification algorithm (Hall and Smith, 1999). The wrapper-based selection was performed simultaneously with classifier selection and we considered three classifiers: LOG, NRBF, and SVM. Wrapper-based methods often achieve better results than the filter-based methods, but are slower as they must repeatedly call the classification algorithm and must be rerun when a different algorithm is used.

2.2.4 Data Mining Software

WEKA

WEKA (Waikato Environment for Knowledge Analysis) is a product of the University of Waikato (New Zealand) and was first implemented in its current form in 1997. WEKA software is written in the Java language and provides a GUI (graphical user interface) for interacting with data files and producing visual results. Weka is open-source software that provides many different algorithms for data mining and machine learning. The software is freely available (http://www.cs.waikato.ac.nz/ml/weka/), platform independent and provides facilities for scripting experiments. It is actively maintained, with selected new algorithms being added as they appear in the research literature.

LIBLINEAR

LIBLINEAR (Library for Large Linear Classification) is a classifier for data with millions of instances and features (Fan et al., 2008). LIBLINEAR is also open source software (http://www.csie.ntu.edu.tw/~cjlin/liblinear/), operates in a scriptable command-line mode, and interfaces with MATLAB/Octave, Java, Python, and Ruby. LIBLINEAR includes implementations for logistic regression and linear support vector machines. It also supports multi-class classification, and cross validation for model selection. We use this software due to the relatively large size of our dataset.

SigmaPlot v12

SigmaPlot is a graph-plotting and curve-fitting package, available online at http://www.sigmaplot.com/downloads. SigmaPlot was used to generate plots and to fit a Gaussian mixture distribution to our data in Chapter 5. It integrates with Microsoft Excel, and provides wizards for plotting and fitting. It also allows for user-defined fitting, presentation of 3-dimensional (3-D) mesh-plots and plots of multiple 3-D plots in the same graph, which were not utilized in this dissertation.

3 Experimental design and evaluation

This chapter summarizes the experimental designs used to develop and validate the models, and also details the performance measures and the statistical tests used in this thesis to evaluate the models.

3.1 Experimental design

We use two experimental designs in our research: the single-split design and the cross-validation design. Both designs generate out-of-sample estimates of the quality of a classifier (predictor). In the single-split protocol, the original data is randomly divided into training and testing datasets. These sets are disjoint, i.e. the original data is sampled without replacement to form both the training and testing datasets. The training dataset is used to inductively determine a prediction model, and the testing dataset is used to determine the out-of-sample quality of that predictor, according to one or more quality metrics. As discussed previously, this is considered a less-biased estimate of the predictor's performance on novel inputs. In the *k*-fold cross validation protocol, the original data is divided into *k* subsets of approximately equal size. A training dataset is formed by merging k-1 of these subsets, while the remaining dataset is treated as the testing dataset. Training and testing process then proceeds as in the single-split design. The entire procedure is repeated k times, with each subset held out as the test dataset exactly once. The quality of the predictions is evaluated by aggregating the out-of-sample performance measures over the *k* iterations. The *k*-fold cross validation method is considered more reliable than the single-split method, as it reduces the risk of randomly

selecting and training and test set on the predictor performs unusually well, which would bias the estimated performance.

3.2 Performance measures

A confusion matrix is commonly used to represent the output of a given classifier (Kohavi and Provost, 1998). Table 3-1 shows a confusion matrix for a binary classification, in which rows represent the observed class labels and columns represent the predicted class labels. The diagonal elements summarize the correctly classified instances and the cross-diagonal elements represent misclassified examples. The entries in the confusion matrix report the number of true positives (*TP*), which are correctly predicted positive examples; false negatives (*FN*), which are the incorrectly predicted positive examples; false positives (*FP*), which are the incorrectly predicted negative examples; and true negatives (*TN*), which are the correctly predicted negative examples.

Table 3-1 Confusion matrix for binary classification.

	Predicted +ve	**Predicted –ve**
Observed +ve	*TP*	*FN*
Observed –ve	*FP*	*TN*

We consider an example in which a classification model predicts a given AA as β-strand or non-β-strand for a sequence with 100 residues that includes 55 β-strand and 45 non-β-strand residues. Among the 55 β-strand type residues, the model predicts 45 as β-strands (*TP*) and 10 as non-β-strands (*FN*). Similarly, among the 45 non-β-strand type residues, 30 were correctly predicted as non β-strands (*TN*) and 15 were incorrectly predicted as β-strands (*FP*). The resulting confusion matrix is shown in Table 3-2.

Table 3-2 Example confusion matrix

	Predicted β-strand	Predicted non-β-strand
Observed β-strand	45	10
Observed non-β-strar	15	30

Next, we define several performance measures computed from the elements of the confusion matrix, which we use to assess our prediction models. These measures quantify prediction quality at the residue and the segment (β-strand) levels. The former means that we assess predictions for each AA, and the latter means that we assess predictions of entire secondary structure segments (in particular, the β-strand segments). These performance measures are consistent with the measures applied in the EVA platform (Koh et al., 2003), which is frequently used to evaluate secondary structure prediction methods.

Residue-level measures include:

Accuracy (***Acc)*** measures the percentage of predictions that are correct and is defined by

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} = \frac{\text{number of correct predictions}}{\text{total number of predictions}} \qquad [3.1]$$

The accuracy measure may not be a sufficient when the number of negative examples is much greater than the number of positive examples in the given data, For example, suppose there are 10000 examples, 9950 of which are negative class and 50 of which are positive class. If the model predicts them all as negative, the accuracy would be 99.5%, even though the classification system misses all positive cases. Hence, there is a need for other measures (Baldi et al., 2000; Rost et al., 2003; Punta et al., 2005; Cheng et al., 2007, etc..) that are at least less sensitive to such a biased class distribution. This is not to say that accuracy is of no value; it is, however, necessary to supplement accuracy with additional quality measures, in order to obtain a complete picture of a classifier's

performance. In the remainder of this section, we present some of the alternatives used in our experiments.

Recall (Q_{ss_obs}) measures the percentage of positive examples that were predicted as positive (i.e. how many of the examples predicted by the model were truly positive among all observed positive examples). This measure is also known as sensitivity and is defined by

$$Q_{ss\_obs} = \frac{TP}{TP + FN} = \frac{\text{number of correct positive predictions}}{\text{total number of positive examples}} \quad [3.2]$$

Precision (Q_{ss_pred}) measures the percentage of positive predictions that are correct (i.e. how many of the examples predicted by the model were truly positive among all predicted positive examples). This measure is defined by

$$Q_{ss\_pred} = \frac{TP}{TP + FP} = \frac{\text{number of correct positive predictions}}{\text{total number of positive predictionss}} \quad [3.3]$$

Eqs. [3.1], [3.2], and [3.3] assume binary prediction where the positive examples are the β-strand residues and the negative examples are the remaining AAs. We also computed Q_{ss} for each secondary structure state separately (i.e. Q_{h_obs}, Q_{h_pred}, Q_{e_obs}, Q_{e_pred}, Q_{c_obs} Q_{c_pred} were also determined).

Per residue prediction accuracy (Q_3) quantifies the three-state per-residue accuracy and is the most widely used score for evaluating secondary structure predictions (Zhang et al., 2010). Q_3 gives the overall percentage of correctly predicted residues in each of the three states: helix, strand and coil. N is the number of residues in a sequence.

$$Q_3 = \frac{100}{N} \sum_{i=\{h,e,c\}} TP_i \quad [3.4]$$

Mathew's Correlation Coefficient (***MCC***) measures the quality of a prediction by comparing the predictions against a random assignment of a class label and is defined by

$$MCC = \frac{((TP * TN) - (FP * FN))}{\sqrt[2]{((TP + FN)(TP + FP)(TN + FN)(TN + FP))}} \qquad [3.5]$$

Matthew's correlation coefficient equals 1 if the prediction is perfect, 0 if the prediction is not better than random and is negative if the prediction is even worse than random (Mathews, 1975).

As in Lin et al. (2005) and McGuffin and Jones (2003), we also computed four quality measures that quantify different types of residue prediction errors based on observed and predicted strand segments; see Figure 3-1 for an example.

(i) ***Over-prediction error*** (***O_e***,) is defined as the number of *FP* residues where the entire segment of the predicted strand residues (predicted β-strand) does not overlap with the observed residues state.

(ii) ***Under-prediction error*** (***U_e***) quantifies the number of *FN* residues where none of the residues in the entire segment of the observed strand residues (observed β-strand) is correctly predicted.

(iii) ***Length error*** (***L_e***) represents the total number of *FN* residues and *FP* residues where some of the predicted strand residues overlap with an observed strand residues and where the incorrect predictions form a segment that extends to a terminus of the observed β-strands.

(iv) ***Inner segment error*** (***W_e***) is defined as the number of *FN* residues which are inside an observed β-strand, i.e., the segment of these incorrect predictions does not extend to a terminus of the observed β-strand.

```
-----EEEE-----EEEE------EEEEEE----EEEE-   observed structure
-EE---EEEE------EEEEE--EE--EE----------   Prediction
-OO--LEEEL----LLEELLL--LEWWEEL----UUUU-   Prediction errors
```

[Over-prediction errors denoted by O, under-prediction errors denoted by U, length error denoted by L, and inner-segment errors denoted by W]

Figure 3-1 Illustration of four types of prediction errors. The top line gives the observed positions of strand residues (E) and non-strand residues (-),the middle line shows a prediction, and the bottom line annotates the errors using bold font.

Segment level measures include:

Segment overlap **(*SOV*$_3$)** quantifies prediction of the secondary structure segments. Given both predicted and observed secondary structure assignments, the segment overlap measure quantifies the overlap of secondary structure segments of each state rather than individual residues (Zemla et al., 1999). A high SOV_3 value means that there is a large overlap between observed and predicted secondary structure segments, and low SOV_3 value indicates smaller overlap. SOV_3 values lie between 0 and 1. This measure has been comprehensively tested during the 2nd critical assessment of techniques for protein structure prediction (CASP2) (Lesk 1997) and the subsequent CASP assessments (these are competitions that assess techniques in the field of protein structure prediction). We computed SOV_3 for three-state secondary structure segments and computed SOV_i for each state separately where i={helix(h), strand(e), and coil(c)} (i.e., SOV_h, SOV_e, SOV_c). We used SOV_e (i.e., for β-strand segments) measure to evaluate the strand prediction model.

Segment overlap for three states *(*SOV_3*)* is calculated as follows:

$$SOV_3 = \frac{1}{N} \sum_{i=\{h,e,c\}} \sum_{s} \frac{\text{minov}(s_{1i}, s_{2i}) + \delta(s_{1i}, s_{2i})}{\text{maxov}(s_{1i}, s_{2i})} \times len(s_{1i}) \qquad [3.7]$$

$$\text{Where } \delta(s_{1i}, s_{2i}) = \min \begin{bmatrix} \text{maxov}(s_{1i}, s_{2i}) - \text{minov}(s_{1i}, s_{2i}); \text{minov}(s_{1i}, s_{2i}); \\ \text{int}(len(s_{1i})/2); \text{int}(len(s_{2i})/2) \end{bmatrix} \qquad [3.8]$$

Segment overlap for β-strand segments *(SOV$_e$)* is calculated as follows:

$$SOV_e = \frac{1}{N}\sum_{s}\frac{\text{minov}(S_{1e}, S_{2e}) + \delta(S_{1e}, S_{2e})}{\text{maxov}(S_{1e}, S_{2e})} \times len(S_{1e}) \qquad [3.9]$$

Where

$$\delta(S_{1e}, S_{2e}) = \min\begin{bmatrix}\text{maxov}(S_{1e}, S_{2e}) - \text{minov}(S_{1e}, S_{2e}); \text{minov}(S_{1e}, S_{2e}); \\ \text{int}(len(S_{1e})/2); \text{int}(len(S_{2e})/2)\end{bmatrix} \qquad [3.10]$$

Where, *N* is the number of residues in a sequence, *s* is the segments, *i* is the secondary structure state, s_{1i} is the observed segment, s_{2i} is the predicted segment, *minov* is the minimum overlap between the observed and predicted segments, *maxov* is the extent of the observed and predicted segments and δ is the accepted variation which assures *minov* over *maxov* ratio to 1.0 where there are only minor deviations at the end of segments. SOV_h and SOV_c is computed similar to SOV_e.

Average strand segments coverage (***ASSC***) is a new measure defined by this thesis that quantifies the overall strand coverage of strands in the sequence. *ASSC* measures how many of the residues are correctly predicted in each overlapping segment pairs from the observed and predicted β-strand segments of a sequence, whether those residues were continuous or not. This differs from SOV_e, which measures the ratio of the *minov* and *maxov* (which are continuous residues of the segment) portions of the overlapping segment pairs from the observed and predicted β-strand segments of a sequence. We used *ASSC* measure to evaluate the prediction model on the overall coverage of strand segments residues in a sequence. This measure is defined by

$$Assc = \frac{\sum_{i=1}^{N} \frac{S_{ip}}{S_{io}}}{N} \qquad [3.11]$$

Where S_{io} is the number of residues in the observed β-strand S_i, S_{ip} is the number of predicted strand residues that overlap with residues in the observed β-strand S_i, and N is the total number of β-strands.

3.3 Statistical significance tests

Our tests of statistical significance for our results proceed in two steps. We first use the Shapiro-Wilk test (Shapiro and Wilk, 1965) to verify normality. The Shapiro–Wilk test tests the null hypothesis that the data in the group forms a normally distributed population. The test statistic is:

$$W = \frac{(\sum_{i=1}^{n} a_i x_{(i)})^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \qquad [3.12]$$

where $x_{(i)}$ is the i-th order statistic, i.e., the i^{th} smallest number in the sample; $\bar{x} = (x_1 + ... + x_n) / n$ is the sample mean and the constants a_i are given by

$$(a_1 .....a_n) = \frac{m^T V^{-1}}{(m^T V^{-1} V^{-1} m)^{1/2}} \qquad [3.13]$$

Where $m = (m_1, ..., m_n)^T$ and m_1, ..., m_n are the expected values of the order statistics of independent and identically-distributed random variables sampled from the standard normal distribution, and V is the covariance matrix of those order statistics. Small values of W indicate non-normality; however, there is no closed-form expression for the distribution of W. Tables of critical values for W may be found in (Pearson and Hartley, 1972).

We used paired t-test (Goulden, 1956) when our data satisfies the normality test. In the paired t-test, the null hypothesis is that the mean difference between pairs $\overline{X}_D$ is zero, while the alternative hypothesis is that the mean difference is nonzero. The test statistic is

$$t = \frac{\overline{X}_D - \mu_0}{s_D / \sqrt{n}} \qquad [3.14]$$

The average ($\overline{X}_D$) and standard deviation (s_D) of those differences are used in the equation. The constant μ_0 is non-zero in a variant that tests the average of the difference is significantly different from μ_0. The *t*-value is then compared to the Student's *t*-distribution with *n*-1 degrees of freedom. The null hypothesis is rejected if and only if the probability $P(t)$ is less than or equal to the chosen significance $\alpha/2$ (for a two-sided test).

We use the Wilcoxon signed rank test (Wilcoxon, 1945) when the data does not pass the normality test. This test analyzes paired differences by using ranks of the data, i.e., it considers the signs of the differences, instead of the magnitude. The null hypothesis is the medians of the two groups are the same, i.e. $m_1 = m_2$, while the alternative is $m_1 \neq m_2$. The test statistic W_+ is defined as

$$W_+ = \sum_{i=1}^{n} \phi_i R_i \qquad [3.15]$$

where R_i is the *i*-th ranked (nonzero) difference between paired items, and ϕ_i is the sign of that difference. W_+ considers only positive differences (W_- considers only negative differences; one uses the smallest of the two as the test statistic). For a small number of differences, an exact computation of the p-value for this statistic is given by (http://www.stat.auckland.ac.nz/~wild/ChanceEnc/Ch10.wilcoxon.pdf). However, for a

larger number of differences (roughly 20 or more), we can approximate the distribution of W_+ as being normal. The z-statistic is

$$z = \frac{w_A - \mu_A}{\sigma_A} \qquad [3.16]$$

$$\mu_A = \frac{n_A(n_A + n_B + 1)}{2} \qquad \text{and} \qquad \sigma_A = \sqrt{\frac{n_A n_B (n_A + n_B + 1)}{12}} \qquad [3.17]$$

w_A denotes the observed rank sum, n_A of our n observations from a distribution are labeled group A and n_B observations from the same distribution are labeled group B. The z statistic is compared against the normal distribution with zero mean and variance = 1

4 Prediction of strand residues

4.1 Overview

The existing approaches for prediction of β-strand residues from protein sequences are characterized by a relatively poor quality. This is likely due to the long-range (in the sequence) interactions that are characteristic of β-sheets, unlike helices and coils. This stems from the fact that extraction of long-range interactions suffers from a combinatorial explosion when compared to more local interactions that can be characterized using one short segment in the chain. Only a few methods have been proposed for the prediction of long-range residue-residue contacts/interactions and their accuracy is relatively low (Hubbard, 1994; Asogawa, 1997; Baldi et al., 2000; Steward and Thornton, 2002; Rost et al., 2003; Punta et al., 2005; Vullo et al., 2006; Cheng et al., 2007). Other reasons for the relatively low predictive performance for the β-strand residues are the weak coupling between β-residues pairs on neighboring strands (Mandel-Gutfreund et al., 2001), i.e., the interactions are "irregular" and thus difficult to find; and the lack of a systematic approach towards the problem (Cheng and Baldi, 2005). At the same time, the knowledge of β-sheet topology, i.e., the pairing of all the β-strands in a given protein, is essential for understanding the structure of β-sheets (Zhang and Kim, 2000). To this end, our work focuses on improving the accuracy of sequence-based prediction of β-strand residues and β-strand segments. Improving these predictions would help with more accurate prediction and understanding of the β-sheet topology and in other areas that were discussed in Section 1.1.

4.2 Existing research and proposed solution

The past three decades have seen intense research in the sequence-based prediction of protein secondary structure (SS) (Rost, 2001). In the last fifteen years, Q_3 for state-of-the-art predictors improved from about 70% (Rost and Sander, 1993) to over 80% (Montgomerie et al., 2006; Zhang et al., 2011). Recent SS predictors employ a variety of machine learning-based models such as neural networks, support vector machines, and regression. They can be categorized into standalone methods and ensembles that combine multiple SS predictors. A majority of the standalone predictors are based on different types of neural networks, including PHD (Rost, 1996), PSIPRED (Jones et al., 1999; McGuffin et al., 2000), SABLE (Adamczak et al., 2005), SSpro (Pollastri et al., 2002), YASPIN (Lin et al., 2005), PORTER (Pollastri and McLysaght, 2005), and SPINE (Ofer and Zhou, 2007). Their Q_3 is relatively high and ranges between 73% and 78% on the benchmark EVA dataset (Rost and Eyrich, 2001; Rost and Sander 2000; Lin et al., 2005). The ensemble predictors include CoDe (Selbig et al., 1999), PROTEUS (Montgomerie et al., 2006; 2008), and CDM (Cheng et al., 2007), and they achieve Q_3 of up to 89.9% on their test datasets (Montgomerie et al., 2008).

The above methods attempt to solve the general three–state prediction problem; however, recent research shows that predicting specific SS types, such as specific coil types including β- and γ-turns (Zheng and Kurgan, 2008; Hu and Li, 2008; Tang et al., 2011) also produces high-quality results. Empirical analysis of two SS predictors, YASPIN and PORTER, reveals that their Q_e values are lower than Q_h by 7 to 16 percentage points (Lin et al., 2005; Pollastri and McLysaght, 2005; Jones, 1999). This indicates that binary classification of strand vs. non-strand residues (either coils or helices) may be characterized by lower improvement over a baseline than the binary classification of

helices or coils (Ward et al., 2003). Furthermore, fragments of protein sequence that fold into strands are characterized by specific patterns that concern occurrence of certain AA types, which were investigated in numerous studies over the last 3 decades (Chou et al., 1982; Chou et al., 1986; Chou and Carlacci, 1991; Mandel-Gutfreund and Gregoret, 2002; Bhattacharjee and Biswas, 2010), and which could be exploited to build effective predictors.

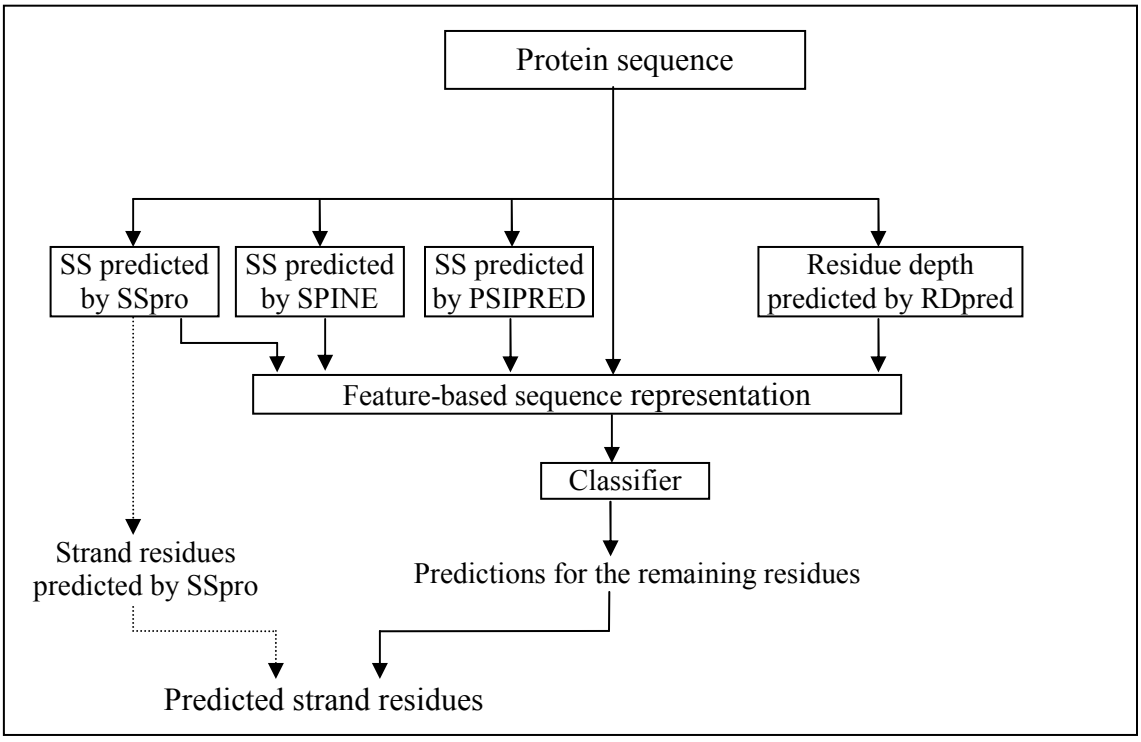
Virtually all modern SS predictors, including PSIPRED, SSPRO, PORTER, and PROTEUS, exploit local information in the sequence using a windowing approach to compute their predictions. In other words, to predict a given AA they use information about the neighboring AAs in the sequence and ignore AAs that are farther away. Their designs also imply independence between positions in the window, i.e. although predictions are based on neighboring AAs they use them individually and do not exploit relations between them. While this is acceptable when considering the AA sequence, windowing the predicted SS sequence (e.g. in the second stage of popular PSIPRED method that uses predicted SS) loses vital information. This recently prompted development of a method that post-processes predicted SS (Madera et al., 2010), and it inspires the development of our feature set. We also note that certain residue characteristics, such as burying depth, that can be predicted relatively accurately from the sequence (Yuan and Wang, 2008; Zhang et al., 2008) and have not been considered by the existing SS predictors, could provide valuable predictive input. More specifically, recent analysis shows that helices are about three times more abundant on the protein surface when compared with strands, while their abundance in the protein core is comparable, and twice as high compared to coils (Yuan and Wang, 2008).

To this end, we propose BETArPRED (BETA Residues PREDictor) (Kedarisetti et al., 2011), which focuses on the two-state strand prediction problem. Our approach is motivated by recent works demonstrating that ensemble-based SS predictors outperform standalone solutions (Montgomerie et al., 2006; 2008; Albrecht et al., 2003). Similarly, combining multiple predictors results in improvements in related predictive efforts, including prediction of protein fold types (Shen and Chou, 2006; Chen and Kurgan, 2007), structural classes (Mizianty and Kurgan, 2009; Kedarisetti et al., 2006; Gromiha and Selvaraj, 1998), quaternary structure type (Shen and Chou, 2006), transmembrane helices (Shen and Chou, 2008), and disorder (Schlessinger et al., 2009; Mizianty et al., 2010; Xue et al., 2010), to name a few. Furthermore, correlations between neighboring SSs are known to be stronger than between neighboring residues (Crooks and Brenner, 2004; Liu et al., 2004). Hence, BETArPRED first passes sequences to three base SS predictors (SSpro, PSIPRED and SPINE). Our selection of these three base predictors is discussed in Section 4.2.3. BETArPRED also uses residue depth predictions computed with RDpred (Zhang et al., 2008) (see section 0), and sequence-derived information to generate features. The features utilize local (window-based) patterns in the predicted SS to exploit relations between adjacent residues. They further combine information about the predicted SS and residue depth, and consider global (sequence-wide) information concerning chain length.

4.2.1 Overview of proposed solution

We propose a novel type of ensemble in which we accept the strand residue predictions of the strongest base method and (re)predict the remaining residues. The overall design of the proposed method is shown in Figure 4-1. The input protein sequence is fed into SSpro, SPINE and PSIPRED to obtain predicted SS. The strand residues predicted by

SSpro are passed to the final prediction due to its best performance on our generic globular protein dataset. This design is motivated by high predictive quality of SSpro for strand residues (as evidenced later in this chapter based on Table 4-1) and the fact that such ensemble provides an improvement in predictive quality over a classical method that would simply combine multiple predictors (as evidenced based on results in Table 4-3 and Table 4-4 which are discussed in Section 4.2.5). The design of the classical method which predicts all residues is denoted on Figure 4-1 when removing the parts shown using dotted lines and it is denoted as alternative design in this thesis. Using our design, the predicted SSs, residue depth predicted with RDpred method (Zhang et al., 2008), and the sequence itself are used to compute a feature vector for the remaining residues that were not predicted by SSpro as strand residues. These features combine both local (window-based) and global (sequence-based) information. The feature vector is passed to a classifier, and the predicted strand residues are merged with the predictions from SSpro.



[the alternative (classical) design can be obtained by removing dotted lines and predicting all residues by a classifier]

Figure 4-1 The overall design of the proposed BETArPRED method.

4.2.2 Datasets

A new dataset is created to train and test the proposed strand prediction model. This dataset has sequences with low (<25%) sequence similarity. The low similarity assures that the solution cannot be found by a simple sequence alignment. This dataset is a subset of PDB (Berman et al., 2000) that are deposited between Jan. 2007 and Dec. 2008; the evaluation was performed in 2009. The main reason to create a new dataset is to make it dissimilar with the previous training sets used by the methods that are used in the proposed ensemble and methods that are compared with the proposed model. These methods were evaluated in 2009 and their corresponding predictive models were built before 2007. Next, this dataset is further filtered to consider only protein chains with high-quality structures; those determined by using X-ray crystallography with resolution < 2.5Å and R-value < 0.25 (Rhodes 2006; Rupp, 2011). Then using CD-hit program (Li et al., 2002), the sequence similarity within the dataset is reduced by selecting a subset of chains that has pair wise sequence identity <40%. Additionally, to minimize the effect of templates used by the methods those are utilized in the proposed solution, we removed any sequence that has >25% similarity to the sequences deposited in PDB before Jan. 2007 using pairwise similarity computed by BLASTp program. As in Cheng and Baldi (2005) and Lippi and Frasconi (2009), we retained the sequences that have at least 50 residues and contain at least 10% strand residues. This is motivated by the fact that metrics that we use to evaluate strand prediction, which are mentioned in Chapter 3, require that strand residues are present in the native structure; otherwise calculations would lead to divide by 0. The final dataset consists of 861 protein sequences, and is referred to as dataset-1 in this thesis. The strand and non-strand residues in this dataset were annotated using DSSP (Kabsch and Sander, 1983).

This dataset was randomly divided into two subsets, the TRAINING and the TEST sets. The TRAINING dataset contains 429 protein sequences (103,390 residues and 25,697 strand residues), which are used to design and train the predictive model using 5-fold cross validation. The TEST dataset contains 432 sequences (106,405 residues and 25,648 strand residues) and is used to determine the out-of-sample prediction quality of BETArPRED. Also, BEATrPRED is evaluated on targets from the CASP8 experiment (Moult et al., 2009). In forming our CASP8 dataset, we exclude 3 targets which could not be processed using DSSP and another 7 for which the predictions of the top-performing tertiary structure predictor in CASP8 (Cozzetto et al., 2009) were missing. This dataset includes 111 sequences (5,358 strand residues, out of 22,875 total residues). The datasets are available at http://biomine.ece.ualberta.ca/BETArPred/BrP.htm.

4.2.3 Empirical evidence on sequence based strand residue and β-strands predictions

We considered the key SS predictors listed by Rost (2009) which include PORTER, PSIPRED, SSpro, SABLE, and YASPIN, as well as two recent predictors, SPINE and PROTEUS2. PSIPRED is widely applied in prediction of various structural properties such as solvent accessibility (Garg et al., 2005), fold (Chen and Kurgan, 2007), structural class (Mizianty and Kurgan, 2009), outer membrane beta barrel protein types (Mizianty and Kurgan, 2011), folding rate (Ivankov and Finkelstein, 2004), and β- and γ-turns (Zheng and Kurgan, 2008; Hu and Li, 2008), to name a few. PROTEUS2 is a recent ensemble method that was selected due to its reported favorable performance when compared with competing SS predictors (Montgomerie et al., 2006; Zhang et al., 2011). YASPIN was reported to provide high quality predictions of strand residues (Lin et al.,

2005; Zhang et al., 2011). PORTER (the standalone version provided at http://distill.ucd.ie/porter/), SSpro 4.0, and SPINE were selected due to their strong performance on the EVA server and in a recent benchmark (Koh et al., 2003; Zhang et al., 2008). We computed *Acc,* Q_{e_obs}, Q_{e_pred}, SOV_3, SOV_h and SOV_e values on the TRAINING set for each of the seven predictors, see Table 4-1. We select the three methods with highest accuracy (SSpro, PSIPRED, and SPINE) to implement our BETArPRED. These methods also have high SOV_e, SOV_h, SOV_3 and Q_{e_pred} values, while their Q_{e_obs} is also relatively large. SABLE and PORTER have low Q_{e_obs}, while PROTEUS and YASPIN over-predict strand residues, leading to low Q_{e_pred}. The selected predictors have $SOV_e < SOV_h$, confirming that helices are predicted more accurately than strands.

Table 4-1 Seven SS predictors compared on the TRAINING dataset.

[The methods are sorted by *Acc*]

SS predictor	*Acc*	Q_{e_obs}	Q_{e_pred}	SOV_e	SOV_h	SOV_3
SSpro	**89.02**	70.49	**82.64**	74.76	80.78	77.33
PSIPRED	**88.71**	73.76	79.24	75.49	80.25	77.51
SPINE	**88.68**	72.01	80.27	75.51	80.02	77.23
SABLE	88.31	68.60	81.29	74.14	78.46	76.48
PROTEUS	87.95	82.42	72.65	78.89	79.24	77.51
PORTER	87.03	66.82	77.74	71.37	78.34	76.08
YASPIN	85.57	72.67	70.18	73.00	76.21	73.41

4.2.4 Features

We employ features generated at three levels: the predicted residue itself (raw values), from a local window centered over the predicted residues (aggregated local information), and the entire protein sequence (aggregated global information). The features are obtained from three input sources: the sequence, the predicted SS, and the predicted depth. In total we extract 214 features for 209795 instances. Out of those 103390 instances used for training and 106405 instances are used for test.

The following feature definitions use the following terminology: the first letter in the prefixes of the feature names indicates the information level, i.e., *r* denotes residue, *w* denotes window, and *p* denote sequence-level features. The second letter of the prefix indicates the information source used to derive the feature value, i.e. *a* denotes to AA information and *s* denotes to SS based information. Additional symbols are defined as follows:

- A method, *i*={PSIPRED, SSPro, SPINE}
- The SS state, *k*={h,e,c}
- SS state segment, a segment formed with a particular SS state
- A depth index, *j*={MSMS, DPX, SADIC}
- Dipeptide Type, *m*={*hh*, *ee*, *cc*, *hc*, *ec*, *ch*, *ce*} We do not consider the dipeptides where strand residues are next to helix residues, since they very rarely occur naturally, if at all.
- Tripeptide Type, *n*={*hhh*, *hcc*, *cch*, *hhc*, *chh*, *hch*, *eee*, *ecc*, *cce*, *cec*, *eec*, *cee*, *ece*, *ccc*, *ech*, *hce*} We do not consider the tripeptides where strand residues are next to helix residues, since they very rarely occur naturally, if at all.
- Fragment Size, *s* = {*3*, *5*, *9*}

Residue-level Features

For each residue in a given sequence, the following 9 features are computed:

r_a_from_N, The linear distance between the N terminal and the current residue position.(1 feature).

r_a_from_C, The linear distance between the C terminal and the current residue position (1 feature).

$r\_a\_ss_{i,}$ The type of SS state predicted by a method (3 features).

r_a_score, The relaibility scores for the predicted SS state by the PSIPRED method (1 feature).

$r\_a\_depth_{,j}$,The depth predicted for a given residue by the RDpred method based on each of the depth indices (3 features).

Window-level Features

174 features are computed for each residue in a given protein sequence using a local window. The maximum window size that we use is 9 (4 residues on each side of the predicted residue). This size was selected since previous work suggests that formation of strands appears to be affected by residues within 3 positions in the sequence (Chen et al., 2006). We extended the resulting 7 residues-wide window to include one more position assuming that feature selection (which is described in the next sub-section) will remove features that are irrelevant.

Among the 174 features, 63 are generated using the predicted residue depths (in some cases combined with the predicted SS):

$w\_a\_depth_j\_frag_s$, is the average depth predicted by a depth index for a given window size (3*3=9 features).

$w\_s\_m_i\_avgdepth\_state_k\_depth_{j,}$ is the average depth predicted by a depth index for each central SS state of a prediction method in a given window. (3*3*3=27 features). A value of -1 is used when a given SS state is not predicted in the window.

$w\_s\_m_i\_avgdepth\_seg_l\_depth_{j,}$ is the average depth predicted by a depth index for each SS state segment of a prediction method in a given window. (3*3*3=27 features). The values for the remaining two SS state segments are set to -1.

Another 87 features quantify composition of the predicted SS:

$w\_s\_m_i\_state_{k,,}$ is the count of SS state residues by a prediction method in a given window (3*3=9 features)

$w\_s\_m_i\_state_k\_norm\_len,$ is the normalized count (by length) of SS state residues by a prediction method in a given window (3*3=9 features).

$w\_s\_m_i\_dipep_{m,}$ is the count of each dipeptide segment type by a prediction method in a given window (7*3=21 features).

$w\_s\_m_i\_tripep_n\_central\_res,$ is the binary feature that describes whether the central tripeptide segment predicted by a method, matches a defined tripeptide segment type in a given window (16*3=48 features).

The following 9 features utilize the reliability scores for the SS predicted by PSIPRED:

$w\_s\_m_{PSIPRED}\_avg\_rel\_score\_state_{k,}$ is the average reliability score for SS state predicted by the PSIPRED method in a given window (3 features).

$w\_s\_m_{PSIPRED}\_max\_rel\_score\_state_{k,}$ is the maximum reliability score for SS state predicted by PSIPRED method in a given window (3 features).

$w\_s\_m_{PSIPRED}\_min\_rel\_score\_state_k$,is the minimal reliability score for SS state predicted by PSIPRED method in a given window (3 features).

The next 9 features quantify the number and size of predicted SS segments:

$w\_s\_m_i\_max\_seg\_len,$ is the maximal length of SS segment predicted by a method in a given window (3 features).

$w\_s\_m_i\_min\_seg\_len,$ is the minimal length of SS segment predicted by a method in a given window (3 features).

$w\_s\_m_i\_seg\_number,$ is the number of SS segments predicted by a method in a given window (3 features).

The final 6 features quantify the position of the predicted residue with respect to the predicted SS segment that includes this residue:

w_s_m_i_max_interface_distance, is the maximal distance between the position of the predicted residue (center of the window) and the two termini of the central SS segment of a method in a given window (3 features)

w_s_m_i_min_interface_distance, is the minimal distance between the position of the predicted residue(center of the window) and the two termini of the central SS segment of a method in a given window (3 features)

Sequence-level Features

A total of 31 features are computed by exploring the entire protein sequence:

p_a_chain_len, is the length of the protein sequence (1 feature).

p_s_m_i_segs, is the number of the SS segments predicted by a method (3 features).

p_s_m_i_seg_l_norm_len, is the count of the SS state segments predicted by a method, normalized by the sequence length (3*3=9 features).

p_s_m_i_seg_l_norm_total, is the count of the SS state segments predicted by a method, normalized by the total SS segments in the sequence (3*3=9 features).

p_s_m_i_Eseg_± 1, is the count of strands where length of the strand predicted by a method equals length of the central segment ± 1 in a given sequence when the central segment is of β-strand type (3 features). These features are set to -1 when the predicted residue in not in a β-strand.

p_s_m_i_Eseg_± 2, is the count of strands predicted by a method where length of the strand equals length of the central segment ± 2 in a given sequence when the central

segment is of β-strand type (3 features). These features are set to -1 when the predicted residue in not in a β-strand.

*p_s_m$_i$_Eseg_* ± *3,* is the count of strands predicted by a method where length of the strand equals length of the central segment ± 3 in a given sequence when the central segment is of β-strand type (3 features). These features are set to -1 when the predicted residue in not in a β-strand.

The *p_s_m$_i$_Eseg_+/-1*, *p_s_m$_i$_Eseg_+/-2* and *p_s_m$_i$_Eseg_+/-3* features help the ensemble when predicting sequences that are rich in β-strands. More specifically, if a residue is in a strand segment, and there are other strands of similar size as this strand in the same protein, the residue is more likely to be a strand since β-strands commonly (although not universally) interact with other strands of similar size to form β-sheets.

The values of -1 are used as feature values in cases where the input information is undefined, e.g., when finding number of similarly sized β-strand for a residue which is not in a β-strand. This value is used by the classifier to identify the fact that value is undefined. We selected -1 since this value is always outside of the domain of these features, e.g., if a given residue is not in a β-strand then the predicate "number of similarly sized β-strand segments" is semantically distinct from the case of a residue in a β-strand where there are no similarly-sized strands in the sequence.

4.2.5 Feature and classifier selection

We use empirical feature selection to identify a subset of features that are effective in predicting strand residues. At the same time, we require a classifier with favorable

predictive quality. These experiments were performed using the WEKA (Hall et al., 2009) and LIBLINEAR (Fan et al., 2008) software packages. We considered two feature selection strategies: a filter-based method in which feature sets are evaluated by their "association" with the prediction outcomes, and a wrapper-based method in which feature sets are assessed based on prediction quality using a given classification method (Hall and Smith, 1999).

We applied two filter-based methods, consistency-based (CONS) (Liu and Setiono, 1996) and correlation-based (CFS) (Hall, 2000) that are described in Section 2.2.3. These two methods were shown to reduce the dimensionality of the feature vector while maintaining or improving prediction quality. Also, these methods were used in other areas of bioinformatics, such as a sequence-based prediction of protein crystallization propensity (Kurgan et al., 2009), prediction of structural classes of proteins (Mizianty and Kurgan, 2009; Kedarisetti et al., 2006), gene expression and protein patterns (Liu and Wong, 2002), where they provided satisfactory results. We used these two selection methods on the TRAINING dataset using 5-fold cross validation and we combined the features selected in each fold together. We also took the union and intersection of these two feature sets, which are denoted UNION and INTER, respectively.

The wrapper-based selection was performed simultaneously with classifier selection. We consider three classifiers that are fast, useful and commonly used in this research context: logistic regression (LOG) (Cessie and Houwelingen, 1992), a normalized gaussian radial basis function (NRBF) network (Bugmann, 1998), and a linear-kernel based Support Vector Machine (SVM) (Fan et al., 2008) due to speed. The RBF network requires setting the number of clusters, k, and we use two variants with k=1 and k=2, referred to as RBF(1) and RBF(2), respectively. These settings allowed for fast calculations given the

large size of our data. We also parameterized the value of complexity constant C for the SVM for each of the feature sets using 5-fold cross validation. As with the filter-based selection, we used best-first search to generate features subsets that were inputted into the four classifiers, LOG, RBF(1), RBF(2), and SVM. Each of the feature sets was evaluated on the TRAINING dataset using 5-fold cross validation. We evaluate the classifiers using three indices: Accuracy (*Acc*), average of Q_{e_pred} and Q_{e_obs} (*Avg*), and SOV_e. Consequently, we have three feature sets.

Next, we used the same four classifiers to compare the predictive quality of all selected feature sets (four selected using filter-based methods and three using the wrapper-based method). Each of the 28 experiments (4 classifiers * 7 feature sets) is based on the 5-fold cross validation on the TRAINING dataset. The complete results are given in Table 4-2. Additionally, we repeated the same procedure with an alternative design (i.e standard ensemble), where all the residues predicted by a classifier. Using results in Table 4-2, we selected four best models based on the highest *Acc* and the highest SOV_e values. Out of those four best models, two models are selected from the standard ensemble, one with highest accuracy and one with highest SOV_e values. Similarly, we selected two best models for the proposed design, one with highest accuracy and one with highest SOV_e. As observed in Table 4-2, two models attain the same highest accuracy for the proposed design and out of those two we chose one best model with the higher Q_{e_pred}.

Table 4-2 Results obtained using 5 fold cross validation on the TRAINING dataset for the considered four classifiers and seven feature sets. Chosen design results shown in bold italics.

[The first two columns specify the classifiers and the feature set identified by the corresponding feature selection method. The results for the proposed design that accepts strand residues predicted by SSpro and predicts the remaining residues are shown in the "(proposed method)" columns. The results for the design that predicts all residues are given in the "alternative design" columns. The results with the highest accuracy (*Acc*) and SOV_e are shown using bold. N/A means that the quality index value could not be computed since no strand residues were predicted.]

Classifier	Feature selection	Prediction of all residues (alternative design)				Strand residues predicted by SSpro with prediction of the remaining residues (proposed method)			
		Acc	SOV_e	Q_{e_obs}	Q_{e_pred}	*Acc*	SOV_e	Q_{e_obs}	Q_{e_pred}
SVM	**CONS**	75.22	0	0	N/A	89.05	74.77	70.53	82.71
	CFS	88.50	73.08	72.13	79.55	89.05	76.21	74.40	80.02
	UNION	75.22	0	0	N/A	89.05	74.77	70.53	82.71
	INTER	88.74	75.51	73.87	79.28	89.05	74.77	70.53	82.71
	ACC	89.07	74.79	70.31	82.98	89.31	76.71	75.40	80.27
	AVG *	89.08	78.75	78.57	77.62	24.78	5.07	100	24.78
	SOVe	55.39	**79.34**	82.73	33.69	55.59	**80.27**	84.64	34.06
LOG	**CONS**	89.25	75.56	72.16	82.26	89.19	77.36	76.16	79.39
	CFS	89.33	74.61	70.91	83.52	89.23	77.23	76.03	79.60
	UNION	89.31	75.09	71.99	82.64	89.24	77.73	76.29	79.45
	INTER	89.02	74.67	69.08	83.77	89.15	76.77	75.69	79.54
	ACC	89.20	75.58	71.92	82.25	***89.51***	78.19	76.63	80.15
	AVG	89.43	75.47	73.15	82.22	***89.51***	78.35	77.00	79.92
	SOVe	88.86	76.49	75.10	78.91	89.45	78.51	77.12	79.66
RBF(1)	**CONS**	88.72	77.24	76.95	77.38	89.23	76.34	74.84	80.36
	CFS	89.37	77.01	75.20	80.61	89.48	78.05	76.64	80.04
	UNION	89.33	76.47	75.45	80.30	89.48	78.05	76.64	80.04
	INTER	89.03	78.31	76.08	78.91	89.35	77.56	75.92	80.07
	ACC	89.07	74.66	70.31	82.97	89.29	77.62	74.32	80.91
	AVG *	**89.54**	77.31	75.62	80.92	89.01	79.51	80.09	76.62
	SOVe	88.74	76.23	76.33	77.82	89.01	79.54	80.05	76.62
RBF(2)	**CONS**	88.89	74.27	72.52	80.69	89.30	76.56	75.07	80.44
	CFS	89.22	75.74	72.30	82.05	89.48	78.05	76.64	80.04
	UNION	89.18	74.31	69.75	83.89	89.48	78.05	76.64	80.04
	INTER	89.07	78.59	76.66	78.69	89.41	78.19	77.20	79.46
	ACC	89.34	77.19	73.43	81.70	89.36	77.66	74.05	81.33
	AVG *	89.00	74.34	73.12	80.67	88.97	79.68	80.86	76.11
	SOVe	88.78	77.30	75.24	78.57	89.16	78.86	78.13	78.12

AVG* means wrapper-based feature selection evaluated using average of Q_{e_pred} and Q_{e_obs}

Table 4-3 compares the four best models against the three base SS predictors on the training dataset. Results show that SSpro has the highest Q_{e_pred} value, i.e., only about 17% of its strand residue predictions are incorrect. This is why strands predicted by this method are passed to the output in our design. However, 29% of the observed strand

residues are missed by SSpro, and our proposed ensemble is designed to find them. Also, Table 4-3 reveals that there are two best results in terms of high accuracy with a tradeoff between Q_{e_pred} and Q_{e_obs}. Out of those two, we selected one best model with highest SOV_e value, which was obtained by the LOG classifier and wrapper-based feature selection evaluated using accuracy (the last row in Table 4-3). The selected feature set used in our best model includes only 9 features and they are discussed in detail later.

Table 4-3 Results of 5-fold cross-validation on the TRAINING dataset for the two best performing feature sets, according to *Accuracy* and SOV_e , using the proposed design and alternative design, they are compared with three base SS predictors (PSIPRED, SSpro and SPINE).

[The proposed/alternative design rows encode the classifiers (SVM, RBF(1), and LOG) and feature selections (SOV_e , *Avg*, and *Acc*) used. The results with the highest accuracy (*Acc*), Q_{e_pred} and SOV_e of the chosen design were shown using bold.]

Predictor		***Acc***	SOV_e	Q_{e_obs}	Q_{e_pred}
SSpro		89.02	74.76	70.49	**82.64**
PSPRED		88.71	75.49	73.76	79.24
SPINE		88.68	75.51	72.01	80.27
Alternative design (predicts all residues)	**SOVe + SVM**	55.39	79.34	82.73	33.69
	Avg + RBF(1)	***89.54***	77.31	75.62	80.92
Proposed design (by taking strand residues predicted by SSpro and predicting the remaining positions)	**SOVe + SVM**	55.59	80.27	84.64	34.06
	Acc + LOG	***89.51***	**78.19**	76.63	80.15

We also compare the four best models from Table 4-3 on the TEST and CASP8 datasets. The results, which are summarized in Table 4-4, confirm that the chosen ensemble provides favorable predictive quality as measured by accuracy, SOV_e and a good trade-off between Q_{e_obs} and Q_{e_pred}. Thus, the proposed BETArPRED method uses the strand residues predicted by SSpro and predicts the remaining residues utilizing the LOG classifier and the 9 features.

Table 4-4 The results on the TEST and CASP8 dataset for the two best performing feature sets, measured by accuracy and SOV_e on the TRAINING dataset, using the proposed design and the alternative design.

[The proposed/alternative design rows encode the classifiers (SVM, RBF(1), and LOG) and feature selections (SOV_e, *Avg*, and *Acc*) used. The results with the highest accuracy (*Acc*) and SOV_e of the chosen design were shown using bold.]

Predictor		**TEST dataset**				**CASP8 dataset**			
		Acc	***SOV_e***	***Q_{e_obs}***	***Q_{e_pred}***	***Acc***	***SOV_e***	***Q_{e_obs}***	***Q_{e_pred}***
Alternative design (predicts all residues)	**SOVe + SVM**	54.57	79.04	82.84	32.64	53.40	76.84	82.42	31.65
	Avg + RBF(1)	***89.42***	78.14	75.29	79.77	***88.10***	70.58	70.64	77.45
Proposed design (by taking strand residues predicted by SSpro and predicting the remaining positions)	**SOVe + SVM**	54.90	79.96	84.48	33.04	53.82	77.66	83.95	32.08
	Acc + LOG	***89.41***	**79.46**	76.60	78.95	***89.70***	**77.65**	76.30	79.62

4.3 Experimental results and discussion

4.3.1 Comparative analysis of predictions of strand residues

Our predictions are assessed using residue level (*Acc,* $Q_{e\_obs,}$ Q_{e_pred}, O_e, U_e, L_e, and W_e) and β-strand segment level (*ASSC* and SOV_e) quality measures. We compare BETArPRED with the seven SS predictors on the TEST and CASP8 datasets. For the CASP8 dataset we also include the best automated 3D structure predictor from the CASP8 experiment (Cozzetto D,,et al.,2009), the ZHANG-server, with the predicted structure processed using DSSP to obtain the positions of strand residues. We include results on the entire CASP8 dataset and also on its two subsets that include sequences with at least 1 strand residue and 10% of strand residues respectively; the 10% amount is consistent with work in (Cheng and Baldi, 2005; Lippi and Frasconi 2009) and with the TEST dataset. This is because most of the quality indices ($Q_{e\_obs,}$ U_e, L_e, W_e, *ASSC*, and SOV_e) could not be measured for chains without strand residues and they may provide statistically unreliable estimates when the number of strand residues is low. In particular, for chains with no strand residues they would default to zero and cannot quantify how

many strand residues are incorrectly predicted. The results are given in Table 4-5. We also assess the statistical significance of improvements on these datasets between BETArPRED and other predictors. This was done by comparing the corresponding results for individual proteins. When a given quality measure for both predictors is normally distributed (per the Shapiro-Wilk test of normality with p-value < 0.05) we applied the paired t-test and otherwise we used the Wilcoxon rank sum test.

Table 4-6 provides these results for different versions of the CASP8 dataset and for the TEST dataset. The results demonstrate that BETArPRED achieves the highest SOV_e and accuracy on the TEST dataset. The *ASSC,* SOV_e, Q_{e_obs}, and U_e of BETArPRED are statistically significantly better at 0.05 when compared with six out of the seven SS predictors. When compared with the remaining PROTEUS which over-predicts strand residues, BETArPRED significantly improves Q_{e_pred}, *Acc,* L_e and W_e. The results on the CASP8 confirm these findings. We note statistically significant improvements in SOV_e, *ASSC,* Q_{e_obs}, and U_e. Overall, the results indicate that BETArPRED accurately predicts individual strand residues (highest accuracy among SS predictors on both CASP8 and TEST sets) as well as β-strands (highest SOV_e, except for YASPIN on the CASP8 set). Importantly, the low values of the U_e, which are significantly lower than most of the other predictors including SSpro, demonstrate that our method finds β-strands that were missed by other methods. When compared with the ZHANG-server, our method significantly improves prediction of strand segments (as measured by *ASSC* and SOV_e) and is inferior in the context of prediction of strand residues. We note that ZHANG-server under-predicts strand residues and these predictions have high quality, while BETArPRED finds substantially more observed strand residues (higher Q_{e_obs}).

Table 4-5 The results of the BETArPRED and the seven representative SS predictors on the TEST and CASP8 datasets, as well as for subsets of the CASP8 datasets that include chains with at least 1 strand residue and at least 10% of strand residues. Results on the CASP8 datasets also include the top-performing automated 3D predictor, ZHANG-server

Dataset	Predictor	*ASSC*	SOV_e	$Q_{e\ obs}$	$Q_{e\ pred}$	*Acc*	O_e	U_e	L_e	W_e
TEST 432 chains with at least 10% strand residues	**BETArPRED**	76.20	**79.46**	76.60	78.95	**89.41**	1.46	2.22	6.89	0.02
	SSPRO	70.58	75.72	71.08	82.03	89.25	1.16	2.86	6.69	0.03
	PSIPRED	72.13	75.53	72.97	77.96	88.49	1.29	3.13	7.07	0.03
	SPINE	70.00	74.87	70.71	79.33	88.48	1.31	2.99	7.17	0.06
	SABLE	67.01	73.37	67.43	79.45	87.92	1.24	3.42	7.41	0.02
	PROTEUS	80.23	77.68	79.99	71.59	87.50	1.35	2.16	8.64	0.09
	PORTER	65.01	70.90	66.17	76.97	87.05	1.47	2.57	7.53	0.02
	YASPIN	73.00	73.16	72.81	68.32	85.28	2.71	3.45	8.55	0.01
CASP8 111 chains	**BETArPRED**	75.75	76.83	75.89	79.96	89.74	1.39	2.28	6.64	0.04
	ZHANG-server	67.49	71.96	67.98	90.30	90.70	0.45	2.87	5.95	0.04
	SSPRO	70.25	73.69	70.65	83.14	89.72	0.93	2.83	6.47	0.04
	PSIPRED	72.02	72.21	72.96	77.26	88.59	1.34	3.13	6.93	0.02
	SPINE	71.43	73.66	71.55	80.13	89.13	1.31	2.93	6.57	0.07
	SABLE	67.27	72.02	67.63	79.88	88.37	1.36	3.37	6.89	0.01
	PROTEUS	74.90	72.00	75.54	72.08	87.36	1.49	3.04	7.99	0.02
	PORTER	63.20	67.44	63.90	75.28	86.57	1.37	3.53	8.51	0.02
	YASPIN	78.88	77.89	79.11	73.14	88.25	1.87	2.57	7.30	0.01
CASP8 106 chains with at least 1 strand residue	**BETArPRED**	75.75	75.74	75.89	79.96	89.33	1.45	2.37	6.90	0.04
	ZHANG-server	67.49	70.63	67.98	90.37	90.34	0.45	2.99	6.18	0.04
	SSPRO	70.25	72.45	70.65	83.14	89.32	0.97	2.94	6.73	0.05
	PSIPRED	72.02	70.90	72.96	77.26	88.13	1.39	3.26	7.20	0.02
	SPINE	71.43	72.42	71.55	80.19	88.72	1.34	3.04	6.83	0.07
	SABLE	67.27	70.70	67.63	80.06	87.96	1.37	3.50	7.16	0.01
	PROTEUS	74.90	70.68	75.54	72.37	86.96	1.55	3.16	8.31	0.02
	PORTER	63.20	65.90	63.90	75.30	86.04	1.42	3.67	8.85	0.02
	YASPIN	78.88	76.84	79.11	73.45	87.89	1.84	2.67	7.59	0.01
CASP8 99 chains with at least 10% strand residues	**BETArPRED**	76.15	80.15	76.23	80.22	89.05	1.43	2.39	7.19	0.04
	ZHANG-server	67.95	72.77	68.35	90.47	90.06	0.46	3.01	6.43	0.04
	SSPRO	70.60	76.63	70.97	83.30	88.99	0.96	2.99	7.01	0.05
	PSIPRED	72.47	75.30	73.37	77.60	87.84	1.34	3.27	7.54	0.02
	SPINE	71.76	76.53	71.86	80.59	88.43	1.28	3.06	7.15	0.07
	SABLE	67.63	75.00	67.95	80.43	87.64	1.32	3.52	7.50	0.01
	PROTEUS	75.35	74.98	75.95	72.70	86.62	1.50	3.17	8.69	0.02
	PORTER	63.51	69.70	64.19	75.26	85.52	1.48	3.73	9.25	0.02
	YASPIN	79.16	80.33	79.38	73.84	87.60	1.77	2.74	7.88	0.01

Table 4-6 The results of the statistical significance tests on the TEST dataset and CASP8 datasets that include 111 chains, 106 chains with at least 1 strand residue, and 99 chains with at least 10% of strand residues which compare BETArPRED against the seven representative SS predictors and ZHANG-server.

[The "---"/"--"/"-" means that BETArPRED is worse with p < 0.02/0.05/0.1, the "+++"/"++"/"+" means that BETArPRED is better with p < 0.02/0.05/0.1, and "=" denotes that the BETArPRED and the other methods are not significantly different]

Dataset	Predictor	*ASSC*	SOV_e	Q_{e_obs}	Q_{e_pred}	*Acc*	O_e	U_e	L_e	W_e
TEST 432 chains with at least 10% strand residues	**SSPRO**	+++	+++	+++	– – –	=	– – –	+++	=	=
	PSIPRED	+++	+++	+++	=	+	– –	+++	=	=
	SPINE	+++	+++	+++	=	+++	=	+++	=	+++
	SABLE	+++	+++	+++	=	+++	=	+++	+	=
	PROTEUS	– – –	=	– – –	+++	+++	=	=	+++	+++
	PORTER	+++	+++	+++	=	+++	– –	++	+	=
	YASPIN	+++	+++	+++	+++	+++	+++	+++	+++	– – –
CASP8 111 chains	**ZHANG-server**	+	=	=	– – –	–	– – –	=	–	=
	SSPRO	+	=	++	=	=	=	=	=	=
	PSIPRED	=	+	=	=	=	=	+++	=	=
	SPINE	=	=	=	=	=	=	++	=	=
	SABLE	+++	++	+++	=	=	=	+++	=	=
	PROTEUS	=	+	=	+++	+++	=	+++	=	=
	PORTER	+++	+++	+++	=	++	=	+++	=	=
	YASPIN	=	=	=	+++	+	=	=	=	–
CASP8 106 chains with at least 1 strand residue	**ZHANG-server**	+	=	=	– – –	– –	– – –	=	–	=
	SSPRO	++	=	++	–	=	=	=	=	=
	PSIPRED	=	+	=	=	=	=	+++	=	=
	SPINE	=	=	+	=	=	=	++	=	=
	SABLE	+++	+++	+++	=	=	=	+++	=	=
	PROTEUS	=	++	=	+++	+++	=	+++	+	=
	PORTER	+++	+++	+++	=	+++	=	+++	=	=
	YASPIN	=	=	=	+++	+	=	=	=	–
CASP8 99 chains with at least 10% strand residues	**ZHANG-server**	++	+	++	– – –	– –	– – –	=	– –	=
	SSPRO	++	+	+++	–	=	=	=	=	=
	PSIPRED	=	++	=	=	=	=	+++	=	=
	SPINE	+	+	+	=	=	=	++	=	=
	SABLE	+++	+++	+++	=	=	=	+++	=	=
	PROTEUS	+++	++	=	+++	+++	=	+++	=	=
	PORTER	+++	+++	+++	+	+++	=	+++	=	=
	YASPIN	=	=	=	+++	+	=	=	=	–

4.3.2 Comparison of 3-state secondary structure predictions

Here, we want to investigate how our two-state BETArPRED predictions compare with existing three-state predictors and how this affects predictions of the other two secondary structure states (helix and coil). To address this, we designed an approach to generate the 3-state predictions using the outputs of the BETArPRED method and we compare them with the considered seven representative SS predictors. By design, BETArPRED predicts all strand residues predicted by SSpro as strands. To generate 3-state predictions for BETArPRED, we combined the outputs of BETArPRED with the predictions from SSpro (which obtains the highest accuracy on the training dataset see Table 4-1), without changing the strand residue prediction assignments of BETArPRED. More specifically, we predict strands for all residues predicted by BETArPRED as strands and all non-strand residues predicted by BETArPRED are assigned the state predicted by SSpro.

We compare these three-state predictions with the corresponding predictions produced by the other secondary structure predictors on the TEST and the CASP8 datasets, see Table 4-7. The results show that the improved prediction of the strand residues provided by BETArPRED does not have a detrimental effect on the prediction of helix and coil residues. The overall three-state predictive quality measured using Q_3 and SOV_3 for BETArPRED is the highest for both datasets. A direct comparison between the 3-state predictions generated by SSpro and the SSpro predictions augmented using the BETArPRED outputs demonstrates that the latter increases both Q_3 and SOV_3 values on the TEST and CASP8 datasets. On the TEST dataset, we observe a small decrease in the SOV_h and SOV_c, and substantially improved SOV_e value.

Table 4-7 Summary of results for the 3-state secondary structure predictions generated by combining predictions of BETArPRED with SSpro and the seven representative SS predictors on the TEST dataset and the CASP8 dataset that includes the automated 3D predictor, ZHANG-server.

[We predict strands for all residues predicted by BETArPRED as strands and we use predictions from SSpro for the non-strand residues predicted by BETArPRED to obtain the 3-state secondary structure predictions]

Dataset	Predictor	Q_3	SOV_3	SOV_h	SOV_e	SOV_c	Q_{h_obs}	Q_{h_pred}	Q_{e_obs}	Q_{e_pred}	Q_{c_obs}	Q_{c_pred}
TEST	**BETArPRED**	**80.45**	**77.56**	81.27	79.46	73.70	84.91	85.30	76.60	78.95	78.85	77.17
	SSPRO	80.12	77.07	81.76	75.72	73.73	85.25	84.56	71.08	82.03	81.04	75.56
	PSIPRED	78.73	76.68	80.25	75.53	72.16	85.47	82.00	72.97	77.96	76.28	76.19
	SPINE	78.82	76.32	80.59	74.87	72.10	84.56	82.89	70.71	79.34	78.65	75.09
	SABLE	77.87	75.90	79.01	73.37	72.40	81.60	84.70	67.43	79.45	80.84	72.04
	PROTEUS	78.48	77.08	78.57	77.68	71.27	84.21	86.40	79.99	71.59	72.58	76.22
	PORTER	76.81	75.50	79.22	70.91	71.91	81.90	82.75	66.17	76.97	78.72	72.04
	YASPIN	75.20	73.56	77.03	73.16	68.56	80.94	81.00	72.81	68.32	71.61	74.48
CASP8 99 chains with at least 10% strand residues	**BETArPRED**	**80.15**	**78.83**	80.16	80.15	73.57	84.61	84.98	76.23	80.22	78.96	76.20
	ZHANG-server	78.14	73.36	78.51	72.77	68.96	85.90	77.42	68.35	90.47	78.19	73.43
	SSPRO	80.10	78.16	81.92	76.63	73.82	85.92	84.23	70.97	83.30	80.99	75.20
	PSIPRED	77.93	77.06	78.88	75.30	71.70	83.81	81.82	73.37	81.82	75.85	74.83
	SPINE	78.71	76.77	78.88	76.53	72.21	83.20	83.16	71.86	80.59	79.29	74.19
	SABLE	77.54	75.94	77.65	75.00	71.60	79.85	85.46	67.95	80.43	81.75	70.74
	PROTEUS	76.38	74.13	73.08	74.98	67.94	81.22	83.27	75.95	72.70	72.57	73.09
	PORTER	74.43	73.29	75.00	69.70	68.80	80.44	80.95	64.19	75.26	75.85	69.00
	YASPIN	78.26	77.39	80.75	80.33	69.99	84.96	82.83	79.38	73.84	71.84	77.25

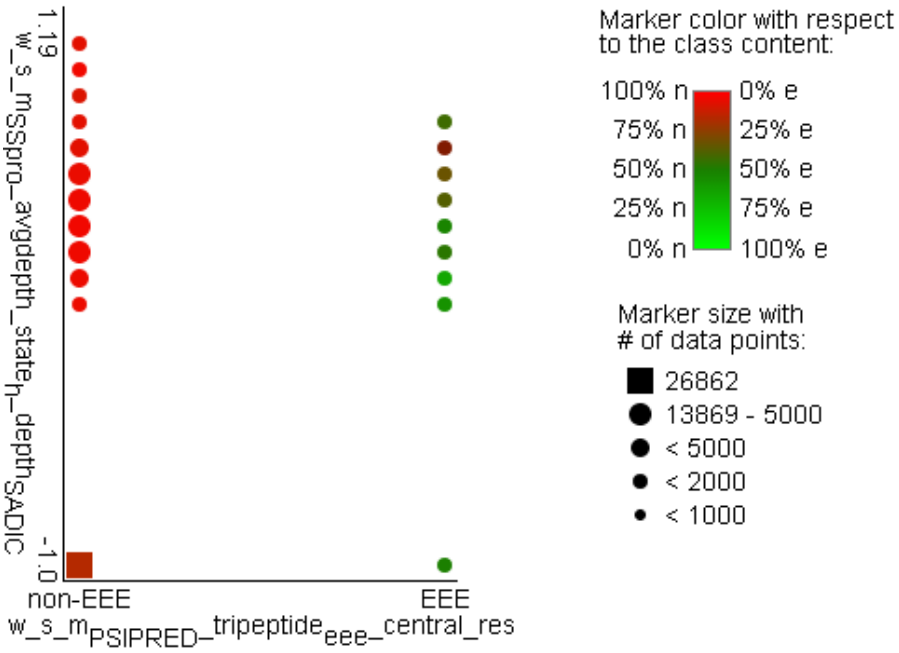
4.3.3 Analysis of the selected features

Error! Reference source not found. lists the features used by BETArPRED. They utilize all considered input predictions at all three information levels. The features use the residue-level SS predicted by PSIPRED and SPINE, local information extracted from the SS predicted by PSIPRED, SPINE and SSpro, a combination of the local predicted SS and residue depth quantified using both volume and distance based definitions, and sequence-level information concerning the chain length. Figure 4-2 visualizes values of two pairs of these features. Both plots show how a given combination of a predicted depth-based feature with a feature that utilizes predicted SS is helpful in annotation of the strand vs. non-strand residues. The position of each marker indicates the values of the two features (essentially the same as a scatter plot, except that each marker represents a cluster of points in this 2D projection). Coloration of the marker indicates the observed

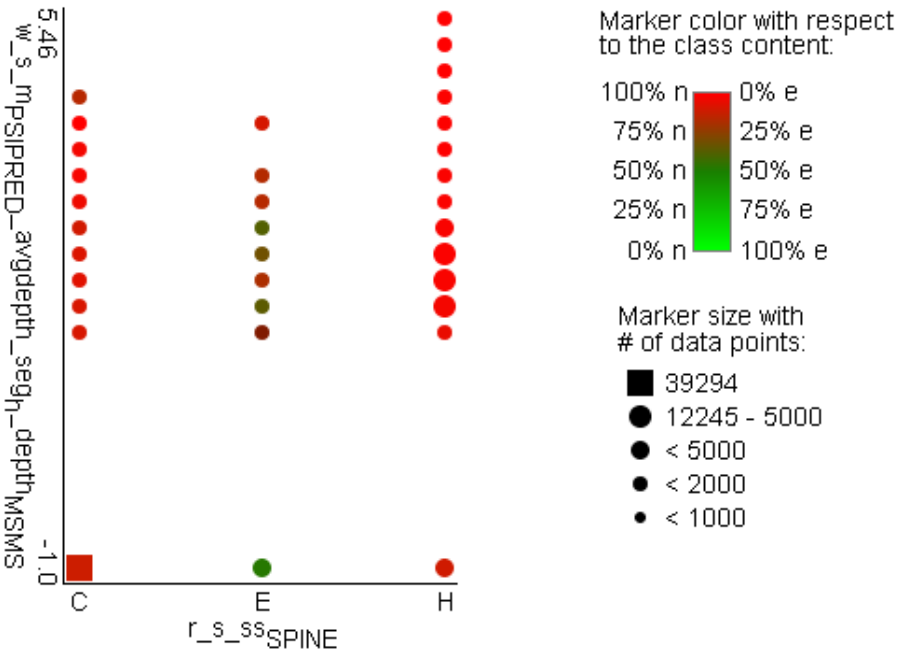
class mixture associated with that cluster of residues, and the shape of the marker indicates the number of residues in the cluster. Note that "predicted SS" that defines features on the *x*-axis comes from a different predictor than for the *y*-axis. When the predicted SS of the residue is a strand (*x*-axis in the bottom panel) or when this residue is located inside a strand segment (*x*-axis in the top panel), the values of the average depth of the predicted helical conformations in the vicinity of this residue (*y*-axes) provide evidence on its proper classification. If there are no predicted helices (-1 on the y-axis), then it is most likely a strand conformation (the marker is green). The higher the average depth of the predicted helices (shown on the *y*-axis), the smaller the likelihood that our prediction should be a strand (the marker is more red). This agrees with the underlying biology, as it is more likely that the predicted helical conformation is correct if its depth is higher.

Table 4-8 Features used by the BETArPRED.

Feature name	Description
r_a_ssPSIPRED r_a_ssSPINE	Residue-level predicted SS by PSIPRED and SPINE
w_s_mPSIPRED_tripepeee_central_res w_s_mPSIPRED_tripepece_central_res w_s_mSPINE_tripepece_central_res w_s_mSSpro_tripepcch_central_res	Local predicted by PSIPRED, SPINE, and SSpro SS of tripeptides, including EEE, ECE, and CCH combinations, centered on the predicted residue
w_s_mPSIPRED_avgdepth_segh_depthMSMS w_s_mSSpro_avgdepth_stateh_depthSADIC	Local average predicted depth of the predicted helix residues and helical segments predicted by PSIPRED and SSpro
p_a_chain_length	Sequence-level chain length



The *y*-axis quantifies the average predicted depth of helical residues predicted by SSpro in a window of size 9 centered on the predicted residue. Value of -1 is used when there are no predicted helices in the window. The *x*-axis shows whether the predicted residue is in the EEE segment, as predicted by PSIPRED.



The y-axis quantifies the average predicted depth for the helix segment predicted by PSIPRED that includes the predicted residues. Value of -1 is used when the predicted residue in not in a helix segment. The x-axis shows th SS predicted by SPINE for the predicted residue.

Figure 4-2 Scatter plots of two pairs of features used by the BETArPRED. Size of the markers denotes number of residues and color denotes their membership (green for strand residues and red for non-strand residues)

4.3.4 Case study

We selected the galactose mutarotase related enzyme Q5FKD7 (PDB ID: 3DCD) among the CASP8 targets to demonstrate our method. This chain contains about 45% strand residues with several short and longer segments. Figure 4-3 shows side-by-side the observed SS derived from DSSP, the results from BETArPRED, and the predictions from the ZHANG-server and SSpro. The results reveal that the proposed predictor finds three β-strands in the middle of the sequence that were missed by SSpro, adding a total of 16 strand residues to the SSpro predictions, out of which 12 are correct and 4 are incorrect. BETArPRED correctly finds additional β-strands as a trade-off for a few over-predicted strand residues located at the termini of the correctly predicted β-strands. An evaluation of the case study predictions is presented in Table 4-9. The ZHANG-server under-predicts the strand residues; only 87 residues were correctly identified, while BETArPRED correctly predicts 111 out of the total of 133 strand residues. The U_e of BETArPRED is 3.7, which is lower by 3.7 and 3.3 when compared with SSpro and ZHANG-server, indicating that our method finds a few extra β-strands. At the same time, this comes as a trade-off for the Q_{e_pred} of BETArPRED that is lower by 2.5 and 8.9 percent when compared with SSpro and ZHANG-server.

```
AA:      XDYTIENNXI KVVISDHGAE IQSVKSAHTD EEFXWQANPE IWGRHAPVLF PIVGRLKNDE
DSSP:    -EEEEE---E EEEEE-B--E EEEEEE---- -B-B------ -------EEB ---------E
SSpro:   --EEEE---E EEEEE----E EEEEE----- EEEEE----- -------EEE E---------E
BrP:     --EEEE---E EEEEE----E EEEEEE---- EEEEE----- -------EEE E---------E
Zhang:   -EEEEE---- EEEEE----E EEEEE----- E--------- -------EE- ----------

AA:      YTYKGKTYHL GQHGFARNAD FEVENHTKES ITFLLKDNEE TRKVYPFKFE FRVNYNLXNN
DSSP:    EEE--EEEE- ---B-----B -EEEEEE--E EEEEEE---- --------EE EEEEEEEE--
SSpro:   EEE--EEEE- ---------- EEEEE----E EEEEEE---- -------EEE EEEEEEE---
BrP:     EEE--EEEE- ---------E EEEEE----E EEEEEEE--- ------EEEE EEEEEEE---
Zhang:   EEE--EEE-- ---------- -EEE------ -EEEE----- --------EE EEEEEEEE--

AA:      LLEENFSVVN KSDETXIFGV GGHPGFNLPT DHGENKEDFY FDXHPSVTRV RIPLKDASLD
DSSP:    EEEEEEEEEE -----EEE-E EE--EEE--- --------EE EEEE----EE E--EE--EE-
SSpro:   EEEEEEEEEE ---------- -----EE--- --------EE E--------- ----------
BrP:     EEEEEEEEEE ------EEE- ----EEE--- --------EE EE-----EEE E---------
Zhang:   EEEEEEEEEE -----EE--- EE--E----- ---------E EEE----E-- ----------

AA:      WNNRSLAPTD SLIALSDDLF KDDALIYELR GNDNKVSLRT DKNKFHVNVW TRDAPFVGIW
DSSP:    ----EEE--- --EE------ ----EEEE-- ---EEEEEEE -----EEEEE EE---EEEEE
SSpro:   ---------- ---------- ---EEEEE-- ----EEEEE- -----EEEEE -----EEEEE
BrP:     ---------- --EE------ ---EEEEE-- ----EEEEEE -----EEEEE E----EEEEE
Zhang:   ---------- --EE------ --EE------ -----EEEE- -----EEEEE E-----EE--

AA:      SQYPKTDNYV CIEPWWGIAD RDDADGDLEH KYGXNHLKPG KEFQAGFSXT YHSTTDEVKL
DSSP:    --------EE EEEEEE---- B-----B--- ----EEE--- -EEEEEEEEE EE--------
SSpro:   --------EE EEE------- ---------- ----EEE--- -EEEEEEEEE EE----EE--
BrP:     --------EE EEE------- ---------- ----EEE--- -EEEEEEEEE EE----EE--
Zhang:   ---------- E--------- ---------- -----EE--- -EEEEEEEEE E---------
```

[where "–" and "E" denote non-strand and strand residues and B denotes beta-bridges, respectively. The AA sequence is split into multiple rows. DSSP is annotated such that bold indicates strand residues missed by SSpro and BrP, and underlined bold shows strand residue segments found by BrP and missed by SSpro. BrP is annotated such that bold / underlined bold indicate mistakes / improvements when compared with SSpro]

Figure 4-3 Comparison of the SSpro, BETArPRED (BrP), and ZHANG-server (ZHANG) predictions with the observed SS derived from DSSP for the galactose mutarotase related enzyme Q5FKD7 (PDBid 3DCD). The DSSP, SSpro, BrP and ZHANG are shown in four consecutive rows.

Table 4-9 The empirical evaluation of the predictions for the case study shown in Figure4-3.

[First column shows the prediction method, second gives the number of correctly predicted strand residues, and next four columns show the strand segment overlap, accuracy for stand residues, and U_e and $Q_{e\ pred}$ measures.]

Method	# Strand residues correctly predicted	SOV_e	Acc	U_e	$Q_{e\ pred}$
SSpro	99	79.8	83.0	7.3	89.2
BETArPRED	111	**86.6**	**85.6**	**3.7**	86.7
ZHANG	87	74.5	80.7	7	95.6

4.4 Summary

BETArPRED is empirically shown to improve predictions of strand residues and strand segments when compared to a wide range of modern SS predictors and with the best-performing tertiary structure predictor in CASP8. It could thus be useful in prediction of higher level structures such as β-sheets (Cheng and Baldi,2007; Max et al., 2010). Since BETArPRED performs well for low identity chains its outputs could be useful in the context of the development of improved sequence profile-profile alignments (Wu and Zhang, 2008). The improvements stem from the novel design, which uses features that aggregate and combine information coming from three SS predictors and the residue depth predictor. Although the BETArPRED provides high quality predictions, there is still room for further improvement. One potential approach could be to exploit strand-strand interactions. This could be done with the help of scoring profiles that reflect inter-strand amino acid pairing preferences, which are tackled in the next Chapter. Another useful source of information that could be used to improve the strand predictions is related to position-specific propensities of AA types in strand segments. BETArPRED is freely available at http://biomine.ece.ualberta.ca/BETArPred/BrP.htm.

5 Strand residue-residue pair propensities

5.1 Overview

Prediction of β-sheets requires an understanding of the interactions between component strands. This is dependent on their constituent amino acids, as cross-strand pairing is influenced by the residue side-chains (Fooks et al., 2006). Several investigations on the pairing of amino acids in β-sheets have been carried out and their prediction quality of contacts between β-strands are roughly 32-35% accurate (Lippi and Frasconi,2009; Tegge et al.,2009; Cheng and Baldi, 2007; Punta and Rost, 2005). In our previous chapter, we developed a model that improves prediction of β-strand residues and β-strands, which are the basic building blocks of β-sheets, when compared with the currently available methods. Our method implements a novel ensemble of secondary structure prediction methods. However, the existing β-strand residue prediction methods, including our BETArPRED (Kedarisetti et al., 2011), are largely based on local interactions, i.e. their inputs are implemented using a sliding window of neighboring residues in the sequence (with the exception of BETArPRED that uses one chain-based input). Hence, in this chapter we investigate whether the use of long-range interactions could provide information that would be useful for prediction of the β-strand residues.

In this context, we propose and empirically analyze scoring functions that quantify the propensity of a given residue to form a β-strand based on propensities of specific residue_residue pairs to interact in β-sheets. We also study the impact of residue conservation and the strand directionality on the quality of these scoring functions. We use the scoring functions to differentiate between strand and non-strand residues, to

determine whether these scoring functions would generate a useful input for the prediction of strand residues.

5.2 Existing research and proposed work

Residue_residue pairing preferences were first studied in parallel and antiparallel β-sheets by Lifson & Sander (1979) on a small dataset of 35 protein chains. They suggested that the resulting residue_residue pair correlations are useful in statistical prediction of protein tertiary structure. Later, residue_residue pair preferences in antiparallel β-sheets and parallel β-sheets were studied separately (Wouters and Curmi, 1995; Hutchinson et al., 1998; Zaremba and Gregoret, 1999; Fooks et al., 2006; Cheng et al., 2007; Zhang et al., 2010). Side chain interactions and residue_residue pair preferences within antiparallel β-sheets were studied in two cases: for pairs whose backbone atoms are hydrogen bonded (H-bonded sites), and for pairs which are not hydrogen bonded (non-H-bonded sites) (Wouters and Curmi, 1995). In addition, an experimental study of the interplay between side chain interactions and the stability of β-sheets observed the greatest stabilization for charge-charge interactions and optimally arranged pairings (i.e., how specific residue_residue pair preferences between strands should align) (Merkel et al., 1999). We discuss some of the findings from these studies later in this chapter when analyzing our results. These investigations demonstrate a weak correlation between pairing preferences and suggest that these interactions are not instrumental in determining the β-sheet structures. Moreover, these works use relatively small datasets which include protein structures that were resolved with low resolution, did not attempt to investigate whether the pairing preferences could be used to identify strand residues, and also normalized the pair preferences using a product of probabilities of individual residues, which may not be a good choice when they would be used to identify strands. (Note that in this field,

normalization does not mean a transform of a feature domain to [0, 1], rather it refers to any process that makes the residue_residue pairs commensurate; in the sense that the bias due to the background frequency of the residues has been corrected.)

In recent quantitative studies, the residue_residue pair preferences were applied to predict certain characteristics of β-strands. The β-sheets were predicted from strand-strand pairs identified in native β-strand segments, by computing strand residue_residue pair probabilities based on contact maps (represent the distance between all possible residue pairs of a 3D protein structure using a binary 2D matrix) (Baldi et al., 2000; Cheng et al., 2007). The drawback of these studies is that they utilized the native β-strand segments. A couple of recent studies concern prediction of strand-strand pair orientation using relative frequency of residue_residue pairs that occur in parallel and antiparallel β-sheets (Zhang et al., 2010; Zhang et al., 2009). In other words, given a pair of native β-strand segments, the authors have proposed a method that decides whether this pair interacts in parallel or antiparallel fashion. In summary, these works did not concern the identification of β-strand residues and they assumed that the native β-strand information is known.

Recently, a few related databases were also developed. One database (ICBS) concerns inter-chain strand-strand interactions only (Dou et al., 2004). Inter-chain interactions occur between protein sequences; in a quaternary structure, strands from different chains can form strand pairs, extending a β-sheet across multiple chains. SheetsPair is another database of amino acid pairs that occurs in β-sheets (Zhang et al., 2007), including both inter-chain and intra-chain interactions. However, this database is not suitable for our investigation, as neither sequence similarity nor structure quality (i.e. resolution and R-value) are recorded in this database.

In contrast to prior research, our work uses a relatively large dataset composed of high quality structures, and we propose a scoring function that quantifies propensity of a given residue to form β-strand. We also use a different type of normalization to generate the residue_residue propensities, which is based on native intra-chain strand residue_residue pairs. To the best of our knowledge, our study is the first to investigate the use of the long range strand-strand interactions within a protein chain to identify β-strand residues. We perform a comprehensive study of strand residue_residue pair propensities for parallel and antiparallel strand pairs, and we further investigate the influence of residue conservation on these residue_residue pairs. Similar to previous works that compute residue_residue pair preferences for H-bonded pairs (Wouters and Curmi, 1995; Hutchinson et al., 1998; Mandel-Gutfreund et al., 2001), we also focus on H-bonded strand residue_residue pairs annotated by DSSP. Furthermore, we utilize these propensities to build novel scoring functions, which are shown to be useful for the identification of strand residues.

5.2.1 Goals

Our aim is to investigate the propensities of the residue_residue pairs in strand-strand contacts that occur within a protein chain. This chapter focuses on third Objective that was discussed in Chapter 1. This investigation requires us to complete the following three tasks:

Task 1: To design and compute strand residue_residue pair propensity tables for parallel and antiparallel β-sheets using a large dataset of high quality protein structures.

Task 2: To study the effect of the residue conservation and the directionality of the strand-strand interactions on the propensities of the strand residue_residue pairs.

Task 3: To construct and evaluate scoring functions that can identify β-strand residues in a given protein sequence.

This work is the first to address Tasks 2 and 3; if successful, our results could be used in a new generation of sequence-based β-strand predictors.

5.2.2 Datasets

We derived two datasets to study the residue_residue pair preferences for parallel and antiparallel strand pairs. The first dataset, called the parallel dataset (PR), includes all H-bonded residue_residue pairs that occur in parallel strand pairs; the second dataset, called the antiparallel dataset (APR), includes all H-bonded residue_residue pairs that occur in antiparallel strand pairs. These two datasets were derived from the 861 protein structures from the dataset described in Section 4.2.2. The strand residue_residue pair details for each protein were determined using DSSP. We separated the strand-strand pairs into parallel and antiparallel subsets based on the orientation of a given pair of strands. The PR dataset contains 2001 strand pairs and 13218 residue_residue pairs. The APR dataset contains 4012 strand pairs and 32008 residue_residue pairs. Note that there are almost twice as may strand pairs in APR dataset compared to PR dataset, which shows that the antiparallel strands are more prevalent than the parallel strands. This is a result of the fact that antiparallel strands are more stable, see Section 2.1.4 for details. To normalize the propensity tables, we computed a third dataset called NR (normalization dataset) based on a random pairing of residues in a given protein chain. We first generated a set of segments extracted from protein chains that follow the distribution of observed strand segment sizes, separately for parallel and antiparallel strands. Next, we aligned each of these segments at a random position in the same protein chain to extract the corresponding residue_residue pairs. Finally, we used these "random" residue_residue

pair propensities to normalize propensities calculated for the parallel or antiparallel native (within the same protein chain) strand residue_residue pairs. We believe that normalizing by the random pairings can better quantify propensities of residue_residue pairs when compared with the normalization using a product of propensities of the two individual residues, particularly in a context of using these propensities to find strand residues within a protein chain.

5.2.3 Propensity scores

Normalization of residue_residue pair scores

Residue_residue pair propensity scores are computed for parallel and antiparallel datasets separately to build two 20x20 dimensional tables. These pairs are symmetric and thus the tables include 210 values (190 heterogeneous pairs that include two different AA types and 20 homogenous pair of a given AA type with itself). To generate these scoring tables, first we count the occurrences of the strand residue_residue pairs in the PR, APR, and NR datasets. Next, we perform normalization by dividing the count of a given strand residue_residue pair in the PR and APR dataset with the count of the corresponding residue_residue pair in the NR dataset. This normalization is quite different from the previous works (Zhang et al., 2010), where a product of probabilities of the two individual residues was used to approximate the probability of that pair. Also, we compute residue pair probability using intra-strand pairs rather than (approximated) inter-strand pairs. An element in the propensity table for parallel and antiparallel direction is computed as follows:

$$m_{ij\_pe} = \frac{P(A_i - A_j)_{pe}}{P(A_i - A_j)_{nd}} \qquad [5.1]$$

$$m_{ij\_ape} = \frac{p(A_i - A_j)_{ape}}{p(A_i - A_j)_{nd}} \qquad [5.2]$$

where the scores m_{ij_pe} and m_{ij_ape} represent propensity of A_i_A_j (residue_residue) pairs that interact in a parallel or antiparallel strand pair , respectively, $P(A_i\_A_j)_{pe}$ and $P(A_i\_A_j)_{ape}$ represent the count of the A_i_A_j residue_residue pairs computed from the PR and APR datasets respectively, and $P(A_i\_A_j)_{nd}$ represents the count of the A_i_A_j residue_residue pairs computed from the NR dataset.

Strand residue_residue pair propensity scores

The strand residue_residue pair propensity scores for parallel and antiparallel datasets are given in Table 5-1 and Table 5-2, respectively. The propensity values >1 indicate the corresponding residue_residue pairs are more likely to form a strand pair for a given direction when compared with a random pairing of residues in the same protein chain. Note that the two matrices are both upper triangular since we only consider 210 possible amino acid pairs, regardless of the arrangement of the two amino acids within a pair (i.e., these matrices are symmetric).

Table 5-1 Strand residue_residue pair propensities that occur in parallel strand pairs calculated based on dataset-1.

[The maximal score is shown in bold font and the underlined scores correspond to pairs that are discussed in (Fooks et al., 2006)]

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	0.726	1.517	0.106	0.211	1.178	0.404	0.650	2.239	0.439	1.127	1.031	0.298	0.140	0.195	0.450	0.437	0.550	2.312	0.723	1.003
C		<u>1.084</u>	0.407	0.674	2.501	0.709	1.530	4.978	0.542	1.264	0.394	0.241	0.056	0.799	0.487	0.785	1.490	3.497	0.619	1.626
D			0.084	0.048	0.173	0.243	0.357	0.335	0.224	0.300	0.193	0.263	0.062	0.137	0.436	0.246	0.473	0.328	0.185	0.185
E				0.070	0.552	0.190	0.774	0.380	0.852	0.355	0.155	0.189	0.104	0.261	0.823	0.435	0.520	0.568	0.514	0.581
F					2.504	1.172	0.933	3.747	0.379	2.272	2.517	0.368	0.099	0.183	0.632	0.461	0.647	4.069	0.820	2.464
G						0.199	0.361	1.240	0.219	0.799	0.585	0.237	0.091	0.207	0.370	0.388	0.627	1.013	0.973	0.593
H							0.929	1.213	0.751	0.624	0.848	0.591	0.000	0.255	0.609	0.661	1.265	1.266	0.401	1.369
I								<u>**13.993**</u>	0.787	5.218	4.286	0.811	0.356	0.656	0.805	1.223	1.455	<u>6.602</u>	2.662	3.252
K									0.117	0.467	0.830	0.310	0.075	0.370	0.511	0.382	0.618	0.935	0.189	0.755
L										<u>2.659</u>	1.781	0.354	0.173	0.420	0.600	0.367	1.001	4.581	1.177	1.982
M											1.239	0.299	0.176	0.271	0.221	1.041	1.699	3.331	1.301	1.626
N												0.694	0.074	0.302	0.527	0.400	1.151	0.817	0.642	0.641
P													***0.000***	***0.000***	0.121	0.078	0.110	0.582	0.264	0.464
Q														0.127	0.654	0.307	0.956	0.743	1.548	0.673
R															0.268	0.624	1.501	1.269	0.723	0.973
S																0.371	0.554	0.795	0.161	0.523
T																	0.949	1.962	0.646	1.053
V																		<u>8.758</u>	2.939	3.422
W																			0.000	1.265
Y																				1.231

Table 5-2 Strand residue_residue pair propensities that occur in antiparallel strand pairs calculated based on dataset-1.

[The maximal score is shown in bold font and the underlined scores correspond to pairs that are discussed (Mandel et al., 2001). The scores in underlines italics correspond to pairs that are cross-referenced with the results in (Wouters and Curmi, 1995).]

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	0.747	0.739	0.251	0.383	1.255	0.497	0.675	1.535	0.326	1.024	0.748	0.269	0.139	0.494	0.530	0.494	0.773	1.858	1.547	1.382
C		*2.686*	0.429	0.358	2.445	0.602	1.316	3.514	0.346	0.932	1.221	0.696	0.367	0.895	0.530	0.556	1.454	2.359	2.813	1.940
D			0.365	0.302	0.566	0.241	0.600	0.334	0.720	0.295	0.219	0.326	0.122	0.516	1.034	0.488	0.767	0.588	0.476	0.395
E				0.355	0.878	0.340	0.895	0.772	*1.300*	0.558	0.735	0.673	0.263	0.876	*1.840*	0.697	1.183	0.872	0.835	1.122
F					*2.948*	1.161	1.106	2.656	0.751	2.240	2.137	0.490	0.716	0.942	0.881	0.876	1.309	2.753	2.081	2.706
G						0.406	0.618	0.965	0.287	0.762	0.639	0.293	0.136	0.285	0.352	0.481	0.653	1.113	1.644	1.271
H							0.810	1.481	0.811	0.689	0.934	0.537	0.413	0.755	0.704	1.244	1.716	1.220	1.127	1.343
I								**5.186**	1.024	2.732	2.307	0.648	0.352	0.970	0.931	1.028	1.087	3.294	3.042	2.739
K									0.476	0.517	0.762	0.409	0.198	0.819	0.620	0.839	1.247	1.050	1.324	1.614
L										1.830	1.410	0.315	0.256	0.655	0.775	0.688	0.746	2.364	2.148	2.072
M											2.046	0.617	0.436	1.063	0.365	0.698	1.331	3.471	0.955	1.886
N												0.591	0.138	0.567	0.589	0.629	*1.204*	0.850	1.028	0.590
P													***0.070***	0.083	0.183	0.183	0.363	0.399	0.786	0.799
Q														0.974	*1.127*	1.106	2.069	0.895	1.790	1.621
R															0.785	0.915	1.498	1.304	1.772	1.900
S																0.823	1.552	1.186	0.812	1.055
T																	2.229	1.596	0.800	1.321
V																		4.362	2.579	2.691
W																			1.902	2.648
Y																				2.686

Impact of residue conservation on the propensity of strand residue_residue pairs

We also compute the propensities of conserved strand residue_residue pairs that occur in parallel and antiparallel strand pairs. To generate the corresponding propensity tables, first we compute conservation scores for all AAs in the considered protein chains. These conservation score values are binarized to differentiate between conserved and non-conserved residues by setting a threshold. Next, we compute the propensities of the conserved strand residue_residue pairs for PR and APR datasets with the normalization using the NR dataset.

Computation of residue conservation

The conservation score of an amino acid that typically mutates is higher compared to an amino acid that typically does not mutate easily (Johansson and Toh, 2010). There are several methods, including entropy–based, variance-based, and matrix score-based, to calculate conservation scores from the multiple sequence alignments (Pei and Grishin, 2001). A recent study suggests that the relative entropy measure leads to more biologically relevant results (Wang and Samudrala, 2006). This measure found applications in several areas, such as identification of protein functional site regions (La and Livesay, 2005), identification of binding sites (Zhang et al., 2010), fold recognition (Chen and Kurgan, 2007), identification of residues determining functional specificity of protein subfamilies (Wang and Samudrala, 2006). Therefore, we chose the relative entropy measure to compute residue conservation.

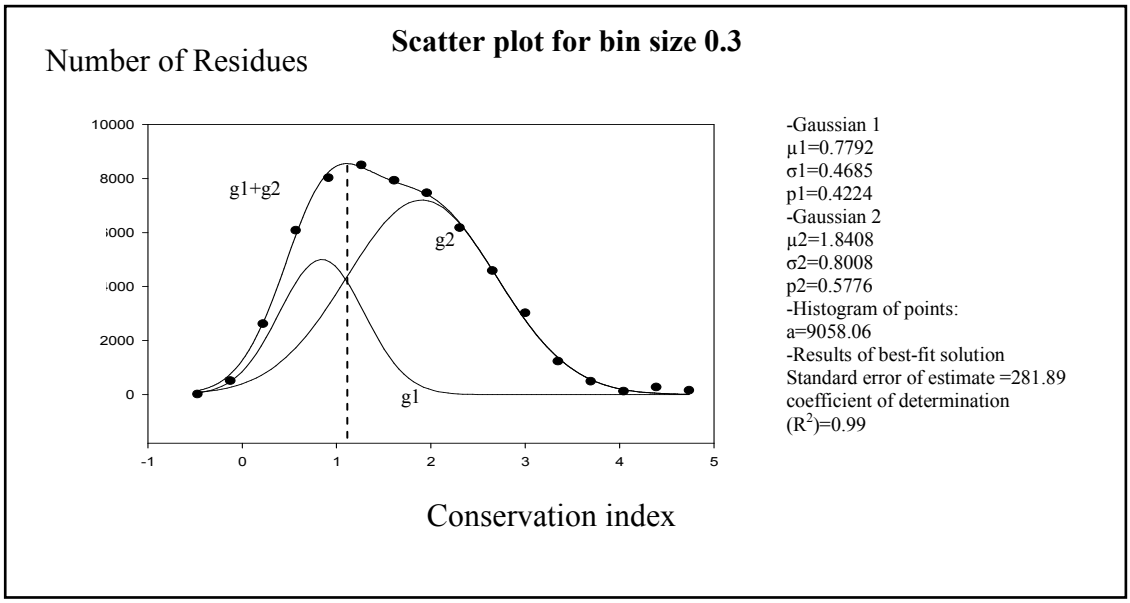
To compute the relative entropy, we first generate PSSM (position specific scoring matrices) profiles for all the protein chains using PSI-BLAST program (Altschul et al., 1990) with default parameters. The relative entropy is defined as follows:

$$RE_j = \sum_{i=1}^{20} WP(A_i) \log \frac{WP(A_i)}{bp(A_i)} \qquad [5.3]$$

where RE_j is the relative entropy of a residue at the j^{th} position in a chain, $WP(A_i)$ are the weighted observed percentages extracted from position-specific scoring matrix (PSSM) for the i^{th} AA type at j^{th} position, and $bp(A_i)$ is the background probability of the i^{th} AA type. The background frequency is the ratio of the occurrence of i^{th} AA type to the total number of residues in the dataset.

We considered residue conservation score values for 51345 strand residues in 861 chains (alignments) from our dataset to establish a threshold that differentiates between conserved and non-conserved strand residues. Based on similar work in (Pei and Grishin, 2001), we plot a histogram of the conservation values; Figure 5-1 presents the histogram with bin size 0.3σ, where σ is the standard deviation. The histogram shows a single peak with asymmetric tails, a short and steep tail on the left side and a long tail on the right side. Such a shape indicates a mixed distribution that is likely to have a few distinct components. This shape is similar to a distribution shown in (Pei and Grishin, 2001), where the authors concluded that it can be approximated by the sum of two Gaussian distributions, one that corresponds to low conservation and another that corresponding to high conservation components. We performed a similar analysis by fitting the sum of two Gaussian distributions with the help of the SigmaPlot software (http://www.ritme.com/tech/sigmaplot); see Figure 5-1. Clearly, the bin size for a histogram has an important influence on the goodness-of-fit for any model fitted to it; we therefore also examined mixture models (again, of two Gaussians) for histograms of the same data at 0.1σ and 0.2σ. We use multiple measures of goodness-of fit: the standard error of the model, the coefficient of determination (R^2) of the model, and a one-way ANOVA (analysis of variance) (see Supplementary Figure 0-1, 2&3 in Appendix A). Overall, while the models for 0.1σ and 0.3σ are nearly as good a fit as each other (and in fact all three are very strong fits), the mixture model for 0.3σ is a slightly better fit than for 0.1σ. This choice is important because if the two distributions correspond to low and high conservation as assumed, then the exact conservation value where these two distributions are equal can be used as a threshold to distinguish low conservation from high conservation. Thus, the threshold depends on our choice of bin size.

Based on the above analysis, and following (Pei and Grishin, 2001), we model the histogram with bin size at 0.3σ (see Figure4.1) as the sum of two Gaussians, which approximate the distribution of the low conservation and high-conservation components, respectively. The low conservation component (on the left) contributes to the main peak together with the left tail. The high conservation component (on the right) gives rise to the long right shoulder. The cut-off threshold to separate conserved and non-conserved AAs is the crossing point of these two Gaussian distributions, equal to 1.0532. This threshold is used to binarize the residue conservation, such that the residue is assumed as conserved if the relative entropy value is > 1.0532, and otherwise it is assumed to be non-conserved.



[The values on the x-axis are binned with the bin size equal 0.3σ where σ is the standard deviation. The corresponding number of residues is shown on the y-axis. Based on (Jimin et al., 2001) and using the Sigmaplot software, these data were fitted into the sum of two Gaussian distributions. (g1 + g2): f=a*(p1*exp(-.5*((x- μ1)/ σ1)2)+p2*exp(-.5*((x- μ2)/ σ2)2)), where μ1 and μ1 are means, σ1 and σ2 are standard deviations and p1 and p2 are coefficients in the sum of two Gaussians. These two Gaussian distributions serve as an approximation of the low conservation and the high conservation components, respectively, and a is parameter that describes bin size*number of residues. We analyzed bin sizes of 0.1 σ, 0.2 σ and 0.3 σ; the best fit to our data (judged by R2 and a one-way ANOVA) occurs for 0.3 σ (see Appendix A for details). The dashed line shows the threshold that is used to binarize the conservation scores]

Figure 5-1 The relative entropy-based conservation score values in histogram, which are shown using black dots.

Conserved strand residue_residue pair propensity tables

Conserved residue_residue pair propensity scores are computed from the PR and APR datasets, respectively, by assuming that both residues in a residue_residue pair should be conserved. There are 4591 and 14406 conserved strand residue_residue pairs from among 13218 and 32008 strand residue_residue pairs in the PR and APR datasets, respectively. An element in the propensity table for parallel and antiparallel direction is computed as follows:

$$mc_{ij\_pe} = \frac{pc(A_i - A_j)_{pe}}{p(A_i - A_j)_{nd}} \qquad [5.4]$$

$$mc_{ij\_ape} = \frac{pc(A_i - A_j)_{ape}}{p(A_i - A_j)_{nd}} \qquad [5.5]$$

where the score $mc_{ij\_pe,}$ mc_{ij_ape} represents the propensity of conserved A_i_A_j- (residue_residue) pairs that interact in parallel or antiparallel strand pairs, repectively, $Pc(A_i\_A_j)_{pe}$ and $Pc(A_i\_A_j)_{ape}$ represents the count of the conserved A_i_A_j pairs computed from the PR and APR datasets respectively, and $P(A_i\_A_j)_{nd}$ represents the count of the A_i-_A_j pairs computed from the NR dataset.

The conserved strand residue_residue pair propensity scores for the PR and APR datasets are shown in Table 5-3 and Table 5-4, respectively. Propensity values >1 indicates that the corresponding residue_residue pairs are more likely to form a strand pairs for a given direction when compared with a random pairing of residues in the same protein chain.

Table 5-3 Conserved strand residue_residue pair propensities that occur in parallel strand pairs calculated based on dataset-1.

[The maximal score is shown in bold font and the underlined scores correspond to pairs that are discussed in (Fooks et al., 2006)]

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	0.499	2.340	0.175	0.101	1.560	0.486	0.965	2.118	0.404	0.815	1.535	0.204	0.108	0.240	0.312	0.210	0.463	1.169	1.040	1.061
C		3.121	0.780	1.386	4.801	1.320	4.406	9.477	0.993	1.734	1.135	0.000	0.000	0.985	0.255	1.399	1.950	2.978	1.783	2.080
D			0.242	0.055	0.200	0.466	0.411	0.255	0.136	0.398	0.555	0.535	0.177	0.225	0.574	0.397	0.574	0.174	0.531	0.246
E				0.122	0.504	0.357	1.465	0.373	0.545	0.335	0.000	0.084	0.187	0.575	0.607	0.339	0.478	0.302	0.423	0.814
F					5.381	1.778	2.340	4.753	0.655	2.837	5.033	0.471	0.284	0.328	0.835	0.620	0.838	3.671	1.518	5.249
G						0.439	0.891	1.749	0.137	0.778	1.189	0.430	0.184	0.278	0.355	0.380	0.592	0.619	1.019	1.290
H							2.675	2.229	0.915	0.943	1.899	0.681	0.000	0.734	1.192	1.294	2.600	1.325	1.156	3.121
I								**16.132**	0.621	5.153	7.348	0.406	0.535	0.524	0.999	1.344	1.824	4.329	4.991	5.305
K									***0.000***	0.392	1.594	0.178	0.043	0.629	0.580	0.065	0.839	0.478	0.000	1.226
L										2.274	1.282	0.295	0.184	0.369	0.261	0.333	1.098	2.485	1.337	2.568
M											3.566	0.861	0.506	0.780	0.510	2.496	2.362	2.588	3.745	4.235
N												1.373	0.107	0.474	1.012	0.443	2.154	0.769	1.618	1.419
P													***0.000***	***0.000***	0.078	0.075	0.317	0.464	0.761	0.817
Q														0.367	1.184	0.236	1.342	0.698	4.161	1.507
R															0.308	0.984	1.640	0.949	1.300	1.274
S																0.450	0.712	0.390	0.462	0.669
T																	1.170	1.327	1.195	1.129
V																		1.676	2.433	3.219
W																			0.000	3.121
Y																				2.193

Table 5-4 Conserved strand residue_residue pair propensities that occur in antiparallel strand pairs calculated based on dataset-1.

[The maximal score is shown in bold font and the underlined scores correspond to pairs that are discussed in (Mandel et al., 2001). The scores in underlines italics correspond to pairs that are cross-referenced with (Wouters et al., 1995)]

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	0.627	0.994	0.244	0.410	1.146	0.569	0.904	1.343	0.306	0.788	0.978	0.286	0.146	0.421	0.436	0.430	0.684	1.058	2.140	1.367
C		*4.972*	0.746	0.442	4.513	1.147	2.808	4.862	0.452	1.243	3.074	0.994	0.663	1.466	1.015	0.823	1.989	1.491	**6.251**	3.149
D			0.483	0.218	0.875	0.355	0.831	0.371	0.778	0.226	0.442	0.440	0.136	0.735	1.029	0.570	0.880	0.420	0.677	0.550
E				0.387	1.248	0.422	1.218	0.638	*1.073*	0.392	0.710	0.641	0.238	0.931	*2.100*	0.516	1.218	0.476	1.180	1.340
F					*5.110*	1.815	2.127	3.688	1.015	2.308	3.721	0.901	1.193	1.423	1.190	1.213	1.722	2.159	4.086	4.136
G						0.559	0.900	1.076	0.227	0.835	1.010	0.377	0.210	0.355	0.386	0.677	0.830	0.851	2.557	1.764
H							1.705	1.799	0.955	0.787	1.730	0.795	0.535	1.248	0.894	1.771	2.122	1.003	2.357	2.670
I								**6.019**	0.831	2.261	2.651	0.663	0.483	1.270	0.909	0.908	1.217	1.870	4.204	3.660
K									0.484	0.423	1.058	0.511	0.192	1.095	0.739	0.845	1.358	0.569	1.989	1.634
L										1.804	1.280	0.299	0.159	0.503	0.697	0.686	0.614	0.997	2.756	2.387
M											4.262	0.823	0.807	2.113	0.487	0.796	1.506	2.571	2.122	3.339
N												1.154	0.153	0.856	0.699	0.776	*1.566*	0.620	1.842	0.836
P													***0.062***	0.144	0.148	0.120	0.332	0.261	1.455	1.231
Q														1.579	*1.509*	1.351	3.101	0.584	2.557	2.607
R															0.835	1.038	1.811	0.788	2.942	3.004
S																*1.147*	2.032	0.812	1.105	1.421
T																	2.507	0.954	1.439	1.496
V																		0.681	2.528	2.039
W																			3.729	4.972
Y																				4.463

5.3 Discussion of propensity scores

5.3.1 Comparison of propensity scores

We plot the strand residue_residue propensity scores for A_i_A_j pairs (in solid black line), conserved A_i_A_j pairs (in dotted line), and the relative frequency A_i_A_j pair scores (in solid grey line) that were recently developed in (Zhang et al., 2010) for parallel and antiparallel pairs in Figure 5-2 and Figure 5-3, respectively. These Figures facilitate a direct comparison of differences between our scores and the recent representative scores that were developed using a different normalization.

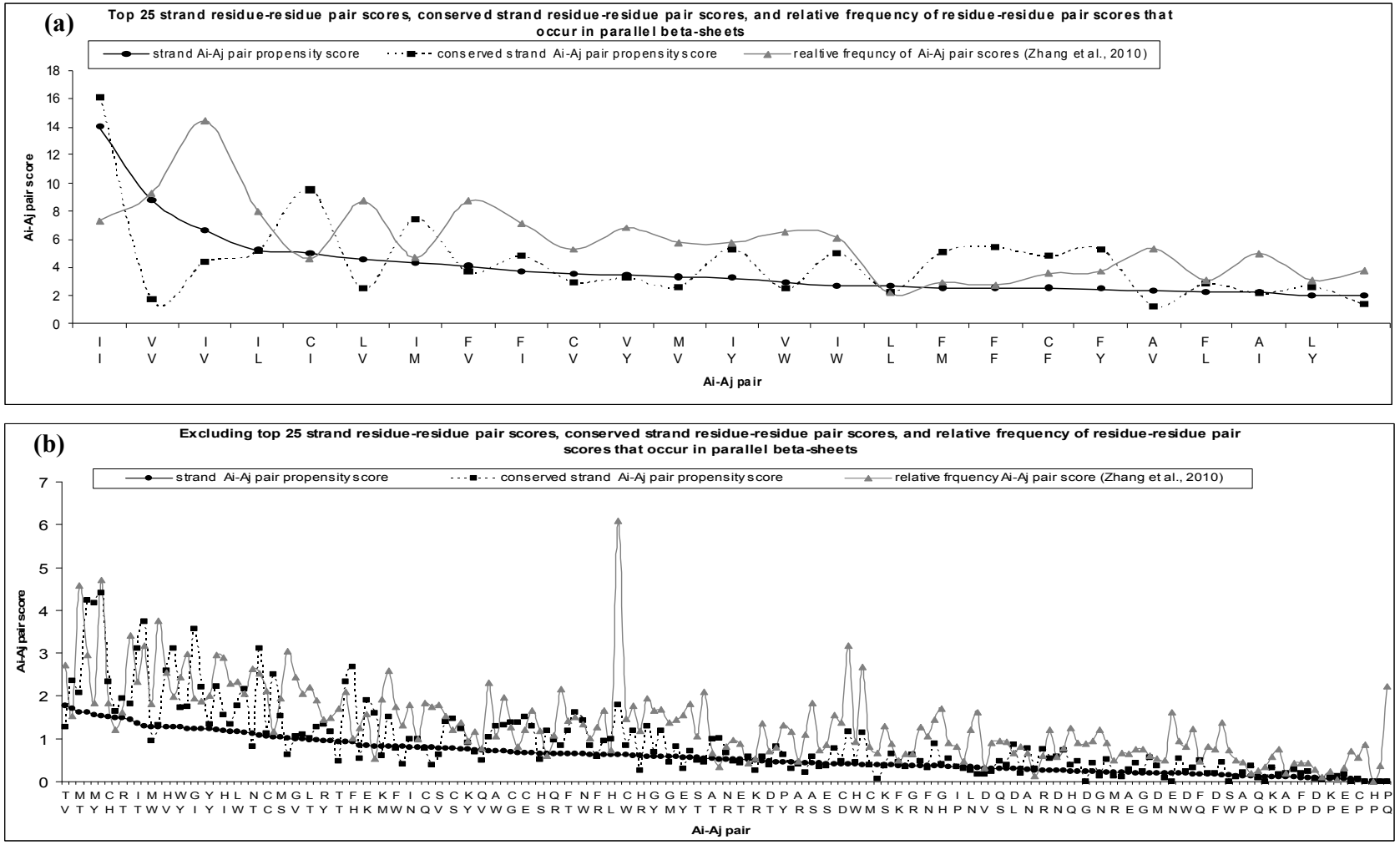


Figure 5-2 210 $A_i\_A_j$ pair scores for parallel direction, where $A_i\_A_j$ pairs are arranged in descending order of strand $A_i\_A_j$ pair propensity scores. Top 25 preferred (with highest values) pairs are shown in panel (a) and remaining pairs shown in panel (b).

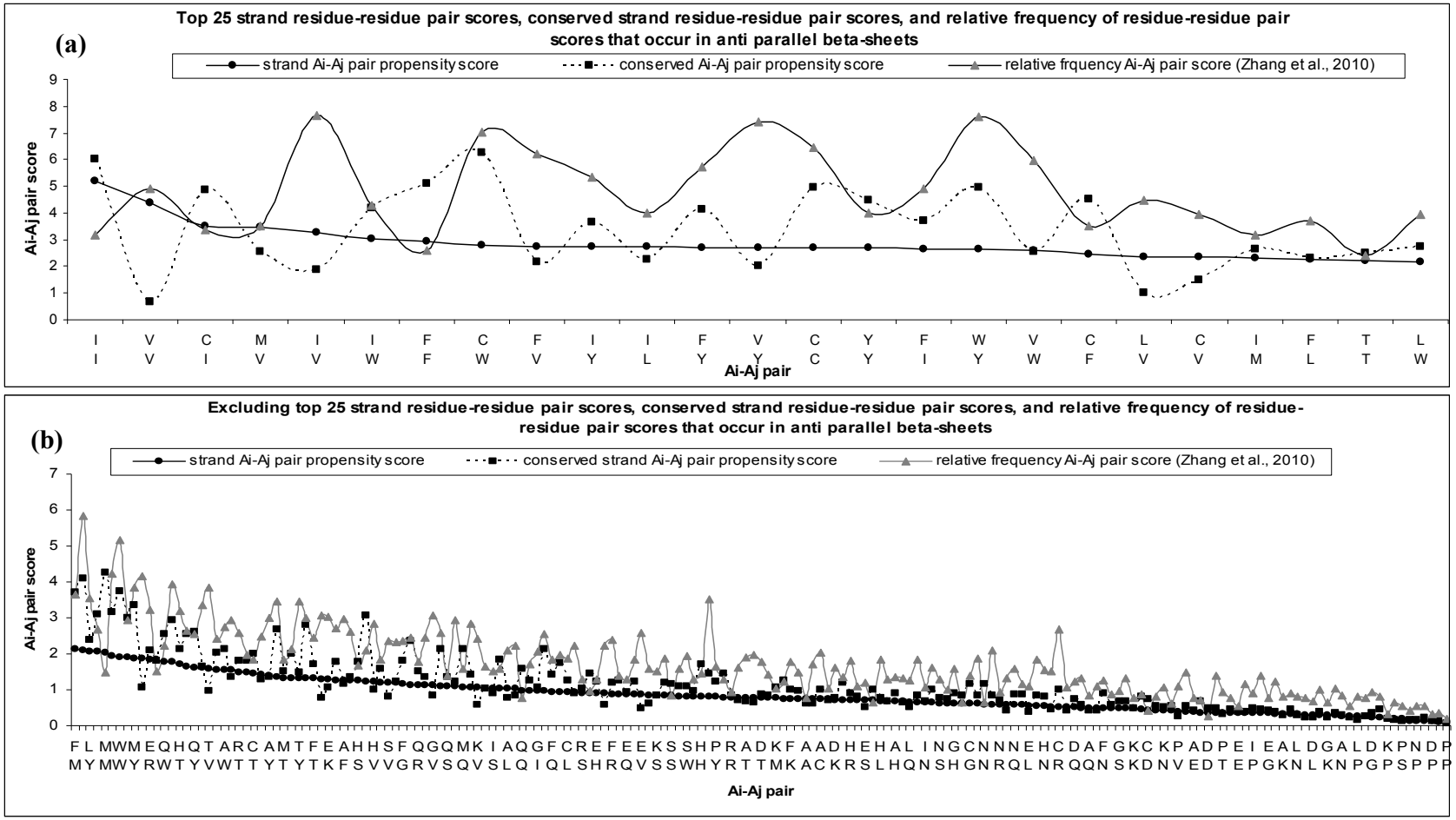


Figure 5-3 210 Ai_Aj pair scores for antiparallel direction, where Ai_Aj pairs are arranged in descending order of strand Ai_Aj pair propensity scores. Top 25 preferred (with highest values) pairs are shown in panel (a) and remaining pairs shown in panel (b)

Based on Figure 5-2 which concerns the parallel direction, among the 210 strand $A_i\_A_j$ pair scores, 28% show preference in formation of strand pairs (i.e., their score value is greater than 1). To compare, the propensities for the conserved pairs and pairs based on work in (Zhang et al., 2010) show preference in 41% and 59% of cases, respectively. For the antiparallel direction, see Figure 5-3, among the 210 strand $A_i\_A_j$ pair scores, 41% have score values > 1. The corresponding percentages for the conserved pairs and pairs based on work in (Zhang et al., 2010) are 50% and 78%, respectively. These results suggest that a larger fraction of residues pairs are preferred for the formation of the antiparallel strands; this could be because the antiparallel strands are more prevalent in protein chains than the parallel strands. Also, a relatively small number of strand $A_i\_A_j$ pairs have high score values (say, >3, which would indicate the three time higher likelihood to form strand pairs), which agrees with other works where only several (<10%) of pairs are show to be significant for the formation of β-sheets (Fooks et al., 2006; Wouters and Curmi, 1995).

Next, we computed Spearman's correlation coefficient between the three sets of strand $A_i\_A_j$ pair scores, for each direction separately. The correlation coefficient values between our propensities for all and the conserved pairs are 0.89 and 0.90 for the parallel and antiparallel direction, respectively, which means these two propensity tables are highly correlated. The coefficient values calculated between our pair scores and the propensities from (Zhang et al., 2010) are 0.83 and 0.89 for the parallel and antiparallel direction, respectively, which again shows that these two approaches to calculate propensity are highly correlated. However, the correlation coefficients between our scores for conserved residues and the scores from (Zhang et al., 2010) are 0.76 and 0.78, respectively, which suggests a lower similarity. This is due to the fact that these scores differ on two aspects: the normalization and the inclusion of the conservation.

Next, we compared the top 5 preferred (highest-scoring) parallel A_i_A_j pairs (I_I, V_V, I_V, I_L, and C_I) from our strand A_i_A_j pair propensity scores with the top 5 preferred pairs from the other two methods, see Figure 5-2. Out of these 5 pairs, 2 (I_I and C_I) and 3 (V_V, I_V, and I_L) also appear among the top 5 based on the conserved strand A_i_A_j pair propensity scores and based on propensities calculated by Zhang et al. (2010), respectively. We also compared the top 5 preferred antiparallel A_i_A_j pairs (I_I, V_V, C_I, M_V, and I_V); see Figure 5-3. Out of these 5 preferred pairs, only 1 pair (I_I) is among the top 5 pairs calculated using the conserved strand A_i_A_j pair propensity scores, and similarly 1 pair (I_V) is among the top 5 based on (Zhang et al., 2010). These results suggest that in spite of the relatively high correlation, the proposed and the existing propensities differ. We also note that based on our scores four of the A_i_A_j pairs (I_I, V_V, C_I, and I_V) are in the top 5 preferred pairs for both parallel and antiparallel directions. Moreover, the A_i_A_j pair with the highest score (I_I) and the A_i_A_j pair with the lowest score (P_P) for parallel and antiparallel directions are the same.

5.3.2 Comparison of propensity scores with existing literature

A recent investigation of A_i_A_j pairs in parallel beta sheets (Fooks et al., 2006), gives a list of nine preferred pairs (N_N, I_I, V_V, L_L, I_V, C_C, D_K, E_K, and E_R). 5 out of these 9 pairs (I_I, V_V, L_L, I_V, and C_S) are among the preferred pairs based on our A_i_A_j pair propensity scores, and 6 pairs (N_N, I_I, V_V, L_L, I_V, and C_C) are preferred based on our conserved A_i_A_j pairs propensity scores. These pairs are shown using underline in Table 5-1 and Table 5-3. An investigation of strand residue pairs for antiparallel β-sheets listed 11 statistically significant A_i_A_j pairs (H_H, G_W, F_L, F_Y, V_Y,V_W, K_S, I_Y, R_S, K_R, and K_Q) that are H-bonded (Mandel et al., 2001). Out

of these 11, 6 pairs are be preferred based on our strand pair propensity scores (G_W, F_L, F_Y, V_Y, V_W, and I_Y), and 8 based on the conserved-pair scores (H_H, G_W, F_L, F_Y, V_Y, V_W, and I_Y); these pairs are shown using underline in Table 5-2 and Table 5-4. In another, older work that analyzes the intra-chain interactions and pair correlations in antiparallel β-sheets, the authors give nine high scoring AA pairs (C_C, E_K, E_R, Q_R, F_F, S_S, D_K, Q_K, and T_N) that are H-bonded (Wouters and Curmi, 1995). Out of the 9, 6 pairs (C_C, E_E, E_R, Q_R, F_F, and T_N) and 7 pairs (C_C, E_K, E_R, Q_R, F_F, S_S, and T_N) are also preferred based on our propensity regular and conservation-based scores, respectively; these pairs are shown using underlined italics in Table 5-2 and Table 5-4. In addition, the ICBS database (Dou et al., 2004) suggests two homo-pairs (the same AAs in a pair): C_C and M_M, and one hetro-pair (different AAs in a pair) C_W, which are favored in the antiparallel β-sheets. These three pairs are also preferred based on our propensities. Our comparative analysis shows that our propensity tables, which are computed, based on the intra-chain normalization, overlap with prior work and that we find a few "new" preferred pairs.

5.3.3 Use of propensity scores to find β-strand residues

Chapter 4 describes the BETArPred method, which improves strand residue prediction when compared with the existing methods. However, there is room left to further improve the strand residue prediction from the sequence. Here, we investigate whether long range interactions, specifically the strand residue_residue pair propensity tables, can be used to differentiate between strand and non-strand residues. To do that, we define a scoring function based on sliding the observed/predicted strand segment over the sequence and aggregating the strand residue_residue pair propensities for the aligned

pairs. We investigate whether these scores differ significantly between the native strand and native non-strand residues.

Scoring function

To compute scoring function values, we first collect all (native or predicted) strand segments in a given sequence. Next, we slide each strand segment in one direction (left to right or parallel, and right to left or antiparallel) along the sequence. While sliding a given segment in a given direction, we compute the scoring function value for each residue in the sequence (for the residues in the center of sliding window), except for the residues that belong to the segment in question; this is to assure that we do not use a given strand segment to "self-score" its own residues. The scoring function is

$$S = \frac{\sum_{n=1}^{L} m_{ij\_n}}{L} \qquad [5.6]$$

where m_{ij} represents the A_i_A_j residue_residue (or conserved residue_residue) pair propensity scores from the corresponding tables (Table 5-1, Table 5-2, Table 5-3, and Table 5-4), depending on the direction of sliding and the use of conservation, and L is the number of pairs that were scored.

The scoring function computes an average strand residue_residue pair propensity over all pairs of residues between the slid strand segment and the corresponding segment on the protein chain. Overlapping segments correspond to a potential matching strand pair, and all AA pairs in these segments can contribute equally to the formation of the strand pair. Higher values of the scoring function means these two segments are more likely to interact to form a strand-strand pair, while lower value means these two segments are less

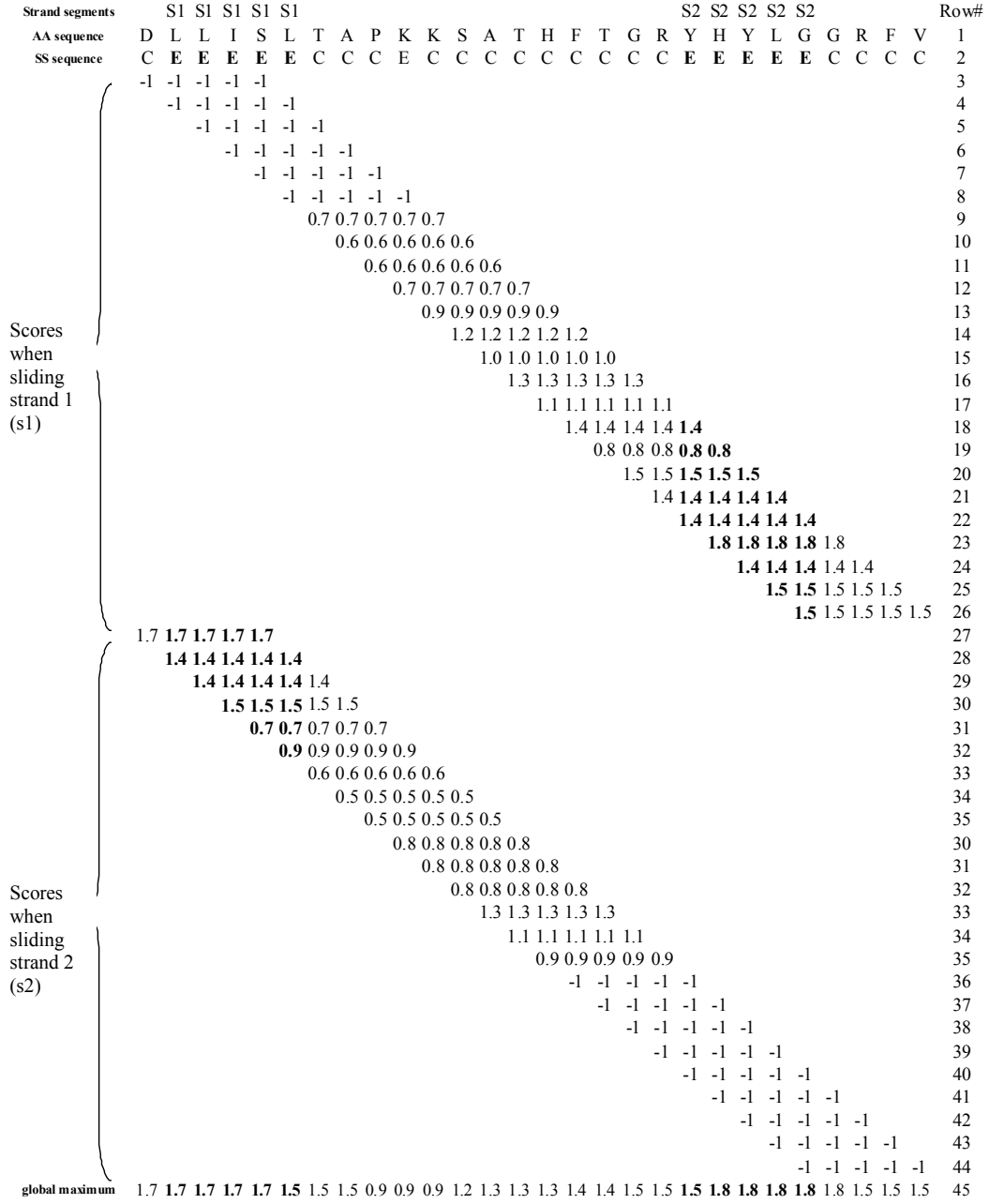
likely to interact. We find the maximum value for each residue among all segments in all sliding positions as follows:

$$Max\_S_i = Max(s_{1i}, s_{2i}, s_{3i}, ...s_{ri}) \qquad [5.7]$$

Where 'i' is the i^{th} residue in the sequence and 'r' is the number of scores available for all the residues. s_{ri} is the score of the i^{th} residue in the r^{th} row.

An example that illustrates the computation of scoring function values for a fragment of the amino acid sequence of the Apolipoprotein A-I Binding Protein (PDB ID: 2DG2) is shown in Figure 5-4. The example includes two strand segments which are slid in parallel direction (left to right). After we calculate the scoring function values for each position when sliding each strand segment, we aggregate these scores by computing a maximum for each amino acid (row 45 in Figure 5-4). We calculate the maximum since each strand must interact with at least one other to form a β-sheet. While most of the strands in a chain *could* interact with more than one other strand, determining the number of interacting strands is *not* our objective. We need only determine that there is a strong interaction with at least one other strand, and this shows that the residue of interest is likely a strand residue. We observe that in our example in Figure 5-4, most of the residues with high maximal scores, at 1.7 and 1.8, do indeed form strand segments. This agrees with the observed structure (see row 2) where these two strands in fact form a β-sheet. We evaluate the quality of these maximum-based propensities through statistical tests that compare their values between native strand and non-strand residues in the next section. We note that we do not calculate the scoring function when the slid segment overlaps with itself in the sequence (rows 3 to 8 for the first strand segment, and rows 36 to 44 for the second strand). Instead, we assign a default value of -1. This value is used since the scoring function values are always positive and we want to ignore these default

values when calculating the maximum. We use this scoring procedure in six cases: (1) when sliding the strand segments in parallel direction; (2) when sliding the strand segments in antiparallel direction; (3) when sliding the strand segments in parallel direction and scoring using only the conserved residue_residue pairs; (4) when sliding the strand segments in antiparallel direction and scoring using only the conserved residue_residue pairs; (5) when sliding the strand segments in parallel direction, scoring using only the conserved residue_residue pairs, and recording the maximal scores only for conserved residues; (6) when sliding the strand segments in antiparallel direction, scoring using only the conserved residue_residue pairs, and recording the maximal scores only for conserved residues. These six cases are evaluated when using (for sliding) both the native strand segments assigned with DSSP and the strand segments predicted with BETArPRED.



[The first row shows the amino acid sequence. The native strand segments, which are identified in row 2 and above the sequence, were assigned with DSSP. Rows 3 to 26 show scoring function values that are obtained by sliding strand 1 segment over the sequence in parallel direction (left to right) and rows 27 to 44 give the scoring function values that are computed by sliding strand 2 segment. Row 45 shows the maximum value calculated for each residue among the corresponding scores generated by all segments in all sliding positions. Scores shown in bold font correspond to location of the native strand segments.]

Figure 5-4 Example computation of the scoring function values for a fragment of the AA sequence of the Apolipoprotein A-I Binding protein (PDB ID: 2DG2).

5.3.4 Evaluation of maximum-based propensities

Table 5-5 summarizes the statistical comparison of the propensities for strand and non-strand residues. We consider the six cases defined in Section 5.3.3, which are shown in consecutive rows in Table 5-5. Specifically, we calculate the maximum-based propensities using all residue pairs and conserved residue pairs by sliding strand segments in the parallel and antiparallel directions, and we calculate the maximum-based propensities only for conserved residues using conserved residue pairs by sliding strand segments in the parallel and antiparallel directions. The above cases were executed when using the DSSP assigned native strand segments, including all segments in both directions (column 3 in Table 5-5) and a subsets of segments in their native direction (column 4 in Table 5-5), and when sliding the BETArPRED predicted strand segments (column 5 in Table 5-5). In case of column 4, we exclude protein chains from which do not have DSSP assigned strand segments in the indicated direction since we cannot score these proteins. As a result, we exclude a total of 348 and 78 sequences from our dataset when assessing propensities for parallel and antiparallel directions, respectively. Similarly for column 5, we exclude 3 protein chains for which BETArPRED did not predict any strand segments. Moreover, when calculating the maximum-based propensities using the conserved residue pairs, when there are no conserved pairs to compute the scoring function then we use the default "-1" value; this is because the scoring function values are always positive and we use maximum to aggregate the scores, so a "-1" value is always ignored.

Table 5-5 Comparison of averages for strand and non-strand residue from dataset-1 in the 18 experimental designs and their statistical significance test results.

[Score types include propensity scores of Table 4.1, Table 4.2 , Table 4.3 and Table 4.4. The directions of sliding include parallel (p) and antiparallel (ap). The tests concern sliding of DSSP assigned native strand segments, including all segments in both directions (column 3), and a subsets of segments in their native direction (column 4), and sliding of BETArPRED predicted segments (column 5). The statistical tests compare scores for native strand residues (E avg) and native non-strand residues (nE avg).Statistical tests are performed by selecting 1000 non-strand and 323 strand residues at random (to maintain the native proportions of strand and non-strand residues) from dataset-1 to compute the averages and to measure significance of the differences between these two sets of scores. This is repeated 1000 times and the "prob. of p-value at 0.05" column reports the probability (fraction) of the 1000 tests where the "E avg" is significantly higher than "nE avg" at the 0.05 level.]

Direc-tion of sli-ding	score type	no direction (all segments) using DSSP-derived strands			direction (only in the correct direction) using DSSP-derived strands			no direction using BETArPRED-predicted strands		
		E avg +/-std	nE avg +/-std	Prob. of p-value at 0.05	E avg +/-std	nE avg +/-std	Prob. of p-value at 0.05	E avg +/-std	nE avg +/-std	Prob. of p-value at 0.05
p	all pairs	3.9514 ±1.6029	3.2355 ±1.3689	1	3.6392 ±1.4531	2.9712 ±1.2039	1	3.5617 ±1.4788	3.3552 ±1.3939	.623
	conserved pairs	5.1484 ±3.7456	4.5320 ±3.4067	.767	4.5842 ±3.4946	4.0270 ±3.1671	.695	4.8252 ±3.5206	4.7151 ±3.4634	.055
	conserved pairs for conserved AAs	5.3521 ±3.4447	4.5463 ±3.0150	.982	4.7949 ±3.2082	4.0997 ±2.7603	.964	4.8541 ±3.3560	4.6345 ±3.2619	.177
ap	all pairs	2.2795 ±0.5395	1.9888 ±0.4660	1	2.0631 ±0.4361	1.8117 ±0.3936	1	2.1182 ±0.5012	2.0383 ±0.4772	.748
	conserved pairs	2.9422 ±1.4553	2.7883 ±1.4390	.396	2.6952 ±1.3907	2.5400 ±1.3847	.476	2.8479 ±1.4200	2.8360 ±1.4314	.019
	conserved pairs for conserved AAs	3.0054 ±1.1816	2.7622 ±1.1315	.936	2.7523 ±1.1096	2.5171 ±1.0740	.952	2.8541 ±1.2969	2.7906 ±1.3086	.124

Table 5-5 shows the average values of the maximum-based propensities over the entire dataset for native strand residues (E avg) and native non-strand (nE avg) residues. We tested the significance of any differences between these values. We could not directly compare the entire set of propensities (51345 and 158450 values for strand and non-strand residues, respectively) due to the large sample size. Instead, we randomly select 1000 values for the non-strand residues and 323 values for strand residues to maintain the native proportions of strand and non-strand residues. Next, we run the normality test for these two sets using Shapiro-Wilk test (Shapiro and Wilk, 1965) at the 0.05 significance

level. Since these two groups were always normally distributed we used the student's group t-test to assess significance. This test was repeated 1000 times, each time selecting a different set of strand and no-strand residues, and we recorded how many times the test was significant at 0.05 significance level. Next we calculated the probability of significance as (sum_plus - sum_minus) / 1000, where "sum_plus" denotes the number of times when the maximum-based propensities of strand residue are significantly higher than for the non-strand residues and "sum_minus" denotes the number of times when the maximum-based propensities of non-strand residue are significantly higher than for the strand residues.

Table 5-5 shows that, as expected, the average maximum-based propensities are always (for all 18 setups) higher for the strand residues. When using the DSSP assigned native strand segments and when scoring using all residues (rows 2 and 5, and columns 3 and 4), the probability of significance is 1, which means that the difference between the scores for strand residues and non-strand residues was always found to be significant. This result suggests that the maximum-based scores are helpful in differentiating strand and non-strand residues. This demonstrates that our propensities can be used to find strand residues based on long-distance interactions.

Table 5-5 also shows that the averages are higher for the parallel direction when compared to the antiparallel direction. For example, when using the DSSP assigned strand segments and when scoring using all residues pairs (rows 2 and 5, and column 3) the average (± standard deviation) for strand and non-strand residues is 3.9514 (±1.6029) and 3.2355 (±1.3689) for the parallel direction, and 2.2795 (±0.5395) and 1.9888 (±0.4660) for the antiparallel direction, respectively. This is due to the higher strand residue_residue pair propensity score values for the parallel direction (see Table 5-1)

when compared to the values for the antiparallel direction (see Table 5-2). Similarly, when we use the conserved strand residue_residue pair scores to compute scoring function values, the averages are higher than when using all strand residue_residue pair scores (e.g., rows 2 and 3, column 3). This is again due to the fact that the residue_residue pair propensity score values for all residues (see Table 5-1 and Table 5-2) are lower than for the conserved pairs (see Table 5-3 and Table 5-4). The above discussion shows that different strand residue_residue pair propensity tables induce different magnitudes of the average means. Therefore, we do not rely on the raw average values to decide whether the maximum-based propensities are useful; instead we use the statistical tests of significance.

The probability of significance when calculating the scoring function using conserved residue pairs (rows 3 and 6) is lower than when using all residues pairs (rows 2 and 5). This is most likely because many segment pairs that are used to calculate the scoring function have no conserved residue_residue pairs, which means that fewer scores are available to calculate the maximums. Similarly, we believe that the probability of significance is lower when we slide the strand segments in their native direction (columns 3 and 4), since again this results in fewer scores that are used to calculate the maximums. Moreover, our analysis in Section 5.3.1 shows that the strand residue_residue propensity scores are similar between the parallel and antiparallel directions, which suggest that directional information does not provide strong predictive value for differentiating strand and non-strand residues.

The last column in Table 5-5 shows the results when we use the predicted strand segments from BETArPRED to calculate the maximum-based scores. As expected, the probability of significance (0.623 and 0.748 for the parallel for antiparallel directions,

respectively) is lower than when using the DSSP assigned segments (column 3). This is because the predicted strand segments may include errors. Nonetheless, this evaluation indicates that our scoring functions can aid prediction of β-strand residues in a protein sequence.

We also compare our propensities (Table 5-1 and Table 5-2) with the propensity scores based on recent work by Zhang *et al.* (2010) (see column 3 & 4 in Table 5-6). For the scores by Zhang *et al.*, the probability of significance is 1 when we slide the observed strand segments (that are assigned by DSSP) in either direction. This is the same as when using the new scores proposed in this work; (see column 3 in Table 5-6).

Table 5-6 Comparison of averages for strand and non-strand residue from our dataset using our scores with propensity scoring tables from (Zhang et al.,2010) for parallel (p) and antiparallel (ap) directions and their statistical significance test results.

[The tests concern sliding of all DSSP assigned native strand segments (column 3) and sliding of BETArPRED predicted segments (column 4). The statistical tests compare scores for native strand residues (E avg) and native non-strand residues (nE avg). We select 1000 non-strand and 323 strand residues at random (to maintain the native proportions of strand and non-strand residues) from our dataset to compute the averages and to measure significance of the differences between these two sets of scores. This is repeated 1000 times and the "prob. of p-values at 0.05" column reports the probability (fraction) of the 1000 tests where the "E avg" is significantly higher than "nE avg" at the 0.05 level.]

Direction of sliding for all pairs	**Residue reside pair scores**	**no direction (all segments) using DSSP-derived strands**			**no direction using BETArPRED-predicted strands**		
		E avg +/-std	**nE avg +/-std**	**Prob. of p-value at 0.05**	**E avg +/-std**	**nE avg +/-std**	**Prob. of p-value at 0.05**
P	**Ours**	3.9514 ±1.6029	3.2355 ±1.3689	1	3.5617 ±1.4788	3.3552 ±1.3939	**0.623**
	others (Zhang et al.,2010)	5.8585 ±1.8143	4.9792 ±1.5342	1	5.3474 ±1.6237	5.1186 ±1.5640	0.611
ap	**Ours**	2.2795 ±0.5395	1.9888 ±0.4660	1	2.1182 ±0.5012	2.0383 ±0.4772	**0.748**
	others (Zhang et al.,2010)	4.1041 ±0.8621	3.6557 ±0.7618	1	3.8592 ±0.8082	3.7361 ±0.7765	0.685

However, when using the predicted strand segments, the probability of significance is higher when using our scores, i.e., 0.623 vs. 0.611 for the parallel direction and 0.748 vs. 0.685 for antiparallel direction (see column 4 in Table 5-6). The 6% improvement in the probability when scanning in the antiparallel direction is relatively substantial, especially considering the fact that the antiparallel strands are more prevalent than the parallel strands. These results suggest that our strand residue_residue propensity tables provide an improvement over the existing approaches in the context of identification of the β-strand residues.

5.4 Summary

In this chapter, we investigated the use of long-range interactions to identify β-strand residues in protein sequences. We computed the propensities of residue pairs to form residue_residue interactions in β-sheets. We also computed the propensities for conserved residue_residue pair interactions that occur in β-sheets to study the effect of residue conservation. We normalized these propensities using residue_residue pair scores in the same chain (intra-chain) rather than normalizing with a product of individual residue scores, as was done in previous works. Our chosen normalization is arguably more suitable for sequence-based prediction of β-strand residues. We also compared these residue_residue propensity scores, including scores based on all and based on conserved pairs, with the relative frequency scores that were recently proposed in (Zhang et al., 2010). Our residue_residue pair propensity scores are shown to be correlated with other two types of scores (scores that include conservation and scores by Zhang et al., (2010)) and our top residue_residue pair propensity scores (pairs that are more prevalent in β-sheets) are in agreement with previously identified residue_residue pairs in β-sheets. However, our empirical evaluation of the utility of these propensity scores in finding β-

strand residues shows that our propensities provide somewhat greater predictive power when compared with the propensities proposed by Zhang et al. (2010).

We also propose a maximum-based scoring function which is used to differentiate between the strand and non-strand residues. These maximum-based scores are based on long-range interactions, and they complement the current β-strand residue predictors that utilize local, window-based information. We assess the usefulness of these scores by comparing their values for strand and non-strand residues. Our results show that the average values for strand residues are significantly higher than for the non-strand residues, meaning these scores can be helpful in differentiating between strand and non-strand residues. We also tested our methods by including the residue conservation and strand directionality information separately and combined, but we were unable to gain further benefits. Overall, our analysis suggests that our residue_residue pair propensities combined with our maximum-based scoring function are potentially useful for the prediction of β-strand residues.

6 Conclusions

6.1 Review

This thesis focuses on improving computational prediction of β-residues (strand residues) and strands in proteins based on the amino acid sequence. Such predictions would lead to better sequence-based recognition of β-sheets. We followed a systematic step-wise approach consisting of three main steps. In the first step, we investigated the two-state performance of existing secondary-structure predictors, and confirmed that Q_e is inferior to Q_h; this indicates that there is a need for a specialized β-strand residue prediction algorithm, to see whether a new prediction method is needed. In the second step, we developed a prediction model that improves prediction of β-residues and strands, when compared with the existing methods. In the third step, we investigated propensities of residue_residue interactions in strand pairs to develop scoring functions that could potentially (in the future) lead to the development of even better strand predictors.

In chapter 4, we developed the BETArPRED model that uses a novel ensemble-based design to predict the β-residues and strand segments. Our BETArPRED predictions are compared with seven modern SS predictors and the top-performing automated structure predictor in CASP8, the ZHANG-server. Our model provides statistically significant improvements over each of the considered SS predictors and improves prediction of β-residues and strands.

In chapter 5, we computed the propensities of residue_residue pairs and conserved residue_residue pairs that interact in β-sheets. We compared the strand residue_residue pair propensity scores with the relative frequency scores of the strand residue-residue

pairs that were recently proposed by Zhang et al., (2010). Our residue_residue pair propensity scores are shown to be correlated with the other two scores (our conservation-based scores and the scores by Zhang et al., (2010)). We also observed that our top ranked residue_residue pair propensity scores (for pairs that are more prevalent in β-sheets) are in agreement with previously published residue_residue pairs in beta-sheets. Next, we proposed a scoring function based on the propensity scores. We found empirically that scores generated by the scoring function can differentiate between strand and non-strand residues. We also observed that inclusion of the residue conservation and strand directionality information does not provide an improvement in differentiating the strand and non-strand residues. Finally, we compared our propensities with the propensities proposed in (Zhang et al., 2010) by using them with the scoring function to differentiate strand and non-strand residues utilizing strand segments predicted by BETArPRED. These empirical results show that our propensities provide greater predictive power for the prediction of β-strand residues.

6.2 Contributions

Objective 1 of this thesis was to compare two-state strand predictions against two-state helix predictions from state-of-the-art secondary structure predictors. Objective 2 was to investigate the creation of a new, more accurate method for the prediction of the strand residues and β-strands. Objective 3 was to investigate the propensities of the residue_residue pairs in the strand–strand contacts, and determine whether these propensities can be used to further improve the prediction of the strand residues and β-strands. My specific contributions for each of these objectives are as follows.

The contributions for the objective 1 are:

- I generated a high quality dataset for prediction of β-residues and β-strands. The criteria applied to compile this dataset were to minimize the effect of templates for secondary structure predictors and to include a generic set of globular proteins that samples from the whole protein structure space with a low sequence similarity and using high quality crystal structures. I divided this dataset into training and test subsets, where the latter subset was used for blind comparison of our predictor that was developed in this dissertation with other relevant predictors. These datasets are available at http://biomine.ece.ualberta.ca/BETArPred/BrP.htm.

- Using my dataset, I empirically investigated how the existing methods perform with respect to β-strands (β-residue) and helix prediction in 2-state rather than 3-state secondary structure prediction. I found, as expected, that prediction of β-strands is characterized by poorer predictive quality when compared with prediction of helices, which motivates investigation of objectives 2 and 3.

My contributions for objective 2 include:

- I introduced an ensemble method (BETArPRED) for prediction of β-strands and β-residues, which utilizes a novel architecture.

- I introduced a new comprehensive set of features that exploits three types of information including the amino acid sequence, predicted secondary structure and predicted residue depth, aggregated at three levels: residue, window and sequence. This set of features is used as an input to BETArPRED.

- I introduced a new quality measure called average strand segment coverage (*ASSC*) to evaluate the strand segment predictions at the sequence level. This measure is different from the currently used *SOV*. The *ASSC* measures how many residues are correctly predicted in a strand segment – whether these residues were contiguous or not, whereas *SOV* calculates the ratio of the largest single contiguous portion of the segment to the largest extended portion of the overlapping segment. If more than one portion of the segment is correctly predicted, those other portions will be ignored. Therefore, *ASSC* complements the *SOV* measure.

- I also evaluated our BETArPRED method using other residue level prediction errors (under, over, length and inner segment errors) in the context of the β-residue predictions and comprehensively compared it with other SS predictors and a leading 3D structure predictor, Zhang-server. My empirical results computed using the test dataset (from objective 1) show that BETArPRED improves predictions of strand residues and strand segments when compared to a wide range of modern SS predictors.

My contributions for objective 3 are:

- I proposed and computed new propensity scores for the strand residue_residue pairs. I compared our residue_residue propensities with other recently proposed propensities (Zhang et al., 2010) and with my propensities calculated using conserved strand residue_residue pairs. I also cross checked my propensities against preferred strand residue_residue pairs identified in the literature. I found that my propensity scores are in agreement with the literature.

- I developed a maximum based scoring function that uses the strand residue_residue pair propensities and I utilized scores generated by this function to differentiate between strand and non-strand residues. Using statistical tests of significance, I found that these scores can distinguish between the strand and non-strand residues.

- I found that the inclusion of sequence conservation and direction of strand segments does not improve the propensity scores in the content of their use for finding strand residues. More specifically, use of the propensity scores that incorporate conservation and direction does not improve the identification of the strand vs. non-strand residues when using my scoring function.
- My empirical analysis demonstrates that our propensity scores provide better discriminative power (to distinguish strand and non-strand residues) when compared with the recent propensities developed by Zhang and colleagues (2010). This was performed by using my scoring function and strand segments predicted by BETArPRED.

6.3 Future work

Although BETArPRED provides high quality predictions, there is still room for further improvement. One potential approach could be to exploit strand-strand interactions. This could be done with the help of our scoring profiles from Chapter 5, and which reflect intra-strand amino acid pairing preferences. (Our results in Table 5-5 are a first step in this direction.) A similar approach was recently proposed and successfully used to predict

relative orientation of a pair of native strand segments (Zhang et al., 2010). In our case, these scoring profiles would be utilized to score the predicted strand residues with respect to their potential match with strand residues on another predicted strand segment. Such an approach would reflect the long range interactions between strand segments that are not covered by the current local window-based strand predictors.

Another useful source of information that could be used to improve the strand predictions is related to position-specific propensities of amino acid types in strand segments. Recent work shows that these propensities are position-specific and that they follow a characteristic periodic pattern in inner positions with respect to the cap residues at both termini of the strand segments (Bhattacharjee and Biswas, 2010). The last extensions of the current method involves flexible windows as proposed by Chou and colleagues (Chou and Shen, 2007; Chou, 2002; Chou, 2001), instead of the fixed-size windows which are used in the current version of BETArPRED, to extract the local information.

References

1. Abdi H, 1994, A neural network primer*, Journal of Biological systems*, Volume 2, Issue 3, Pages 247-283.
2. Adamczak R, Porollo A, and Meller J, 2005, Combining prediction of secondary structure and solvent accessibility in proteins, *Proteins*, Volume 59, Pages 467-475.
3. Anfinsen CB, 1973, Principles that govern the folding of protein chains, *Science,* Volume 181, Isuue 4096, Pages 223–230.
4. Albrecht M, Tosatto SC, Lengauer T, and Valle G, 2003, Simple consensus procedures are effective and sufficient in secondary structure prediction, *Protein Engineering*, Volume 16, Issue 7, Pages 459-462.
5. Allmuallim H and Deitterich TG, 1991, Learning with many irrelevant features, *Proceedings of the Ninth National. Conference on AI*, Pages 47-52.
6. Altschul SF and Lipman DJ, 1990, Protein database searches for multiple alignments, *Proceedings of National Academic Science,* Volume 87, Issue 14, Pages 5509-5513.
7. Altschul SF, Madden TL, and Schäffer AA, 1997, Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, *Nucleic Acids Research*, Volume 25, Issue 17, Pages 3389–3402.
8. Aravind L, and Koonin EV, 1999, Gleaning non-trivial structural, functional and evolutionary information about proteins by iterative database searches, *Journal of Molecular Biology*, Volume 287, Issue 5, Pages 1023–1040.
9. Asogawa M, 1997, Beta-sheet prediction using inter-strand residue pairs and refinement with hopfield neural network, In Proceedings of *International Conference on Intelligent System of Molecular Biology*, Volume 5, Pages 48–51.

10. Baldi P, Pollastri G, Andersen CAF, and Brunak S, 2000, Matching protein β-sheet partners by feed forward and recurrent neural networks, *In Proceedings of the Conference on Intelligent Systems for Molecular Biology*, Pages 25–36.
11. Baldi P and Brunak S, 2001, *Bioinformatics: The Machine Learning Approach,* 2nd edition, MIT Press.
12. Berg JM, Tymoczko JL, and Stryer L, 2002, *Biochemistry*, 5th edition, W.H. Freeman and company.
13. Bhattacharjee N and Biswas P, 2010, Position-specific propensities of amino acids in the β-strand, *BMC Structural Biology*, Volume 10, Article No. 29.
14. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, and Bourne PE, 2000, The protein data bank, *Nucleic Acids Research*, Volume 28, Issue 1, Pages 235-242.
15. Bowie JU, Luethy R, and Eisenberg D, 1991, A method to identify protein sequences that fold into a known three-dimensional structure, *Science*, Volume 253, Issue 5016, Pages 164-170.
16. Branden C and Tooze J, 1999, *Introduction to protein structure*, 2nd edition, Garland.
17. Bugmann G, 1998, Normalized Gaussian Radial Basis Function networks, *Neurocomputing,* Volume 20, Pages 97-110.
18. Cai X,Hu H and Li X, 2009, *A new measurement of sequence conservation*, BMC Genomics, Volume 10, Article No.623.
19. Campbell C, 2002, Kernel Methods: A Survey of Current Techniques, *Neurocomputing*, Volume 48, Pages 63–84.
20. Cessie S and Houwelingen J, 1992, Ridge estimators in logistic regression, *Applied Statistics,* Volume 41, Issue 1, Pages 191-201.

21. Chen K, Kurgan L and Ruan J, 2006, Optmization of the sliding window size for protein structure prediction, *Proceedings 2006 IEEE CIBCB Symposium*, Pages 366-372.

22. Chen K and Kurgan LA, 2007, PFRES: Protein Fold Classification by Using Evolutionary Information and Predicted Secondary Structure, *Bioinformatics*, Volume 23, Issue 21, Pages 2843-2850

23. Cheng J and Baldi P, 2005, Three–stage prediction of protein beta-sheets by neural networks, alignments and graph algorithms, *Bioinformatics,* Volume 21, Pages 75-84.

24. Cheng J and Baldi P, 2007, Improved residue contact prediction using support vector machines and a large feature set, *BMC Bioinformatics*, Volume 8, Article No. 113.

25. Cheng H, Sen TZ, Jernigan RL, and Kloczkowski A, 2007, Consensus data mining (CDM) protein secondary structure prediction server: combining GOR V and fragment database mining, *Bioinformatics*, Volume 19, Pages 2628-2630.

26. Chou KC, Pottle M, Nemethy G, Ueda Y, and Scheraga HA, 1982, Structure of beta-sheets: Origin of the right-handed twist and of the increased stability of antiparallel over parallel sheets, *Journal of Molecular Biology*, Volume 162, Pages 89–112.

27. Chou KC, Nemethy G, Rumsey S, Tuttle RW, and Scheraga HA, 1986, Interactions between two beta-sheets: Energetics of beta/beta packing in proteins, *Journal of Molecular Biology*, Volume 188, Pages 641– 649.

28. Chou KC and Carlacci L, 1991, Energetic approach to the folding of alpha/beta barrels, *Proteins*, Volume 9, Pages 280–295.

29. Chou KC, 2001, Using subsite coupling to predict signal peptides, *Protein Engineering*, Volume 14, Pages 75-79.

30. Chou KC, 2002, Prediction of protein signal sequences. *Current Protein Peptide Science*, Volume 3, Pages 615-622.
31. Chou KC and Shen HB, 2007, Signal-CF: a subsite-coupled and window-fusing approach for predicting signal peptides, *Biochemical Biophysics Research Communication,* Volume 357, Pages 633-640.
32. Cozzetto D, Kryshtafovych A, Fidelis K, Moult J, Rost B and Tramontano A, 2009, Evaluation of template-based models in CASP8 with standard measures, *Proteins,* Volume 77, Supplement 9, Pages 18-28.
33. Cristianini N and Shawe-Taylor J, 2000, *An Introduction to Support Vector Machines*, Cambridge University Press, UK.
34. Crooks GE and Brenner SE, 2004, Protein secondary structure: entropy, correlations and prediction, *Bioinformatics*, Volume 20, Pages 1603-1611.
35. Dou Y, Baisnée PF, Pollastri G, Pécout Y, Nowick J and Baldi P, 2004, ICBS: a database of interactions between protein chains mediated by beta-sheet formation, *Bioinformatics*, Volume 20, Issue 16, Pages 2767-2777.
36. Espadaler J, Romero-Isart O, Jackson RM, and Oliva B, 2005, Prediction of protein-protein interactions using distant conservation of sequence patterns and structure relationships. *Bioinformatics*, Volume 21, Pages 3360–3368.
37. Fan RE, Chang KW, Hsieh CJ, Wang XR, and Lin CJ, 2008, LIBLINEAR: A library for large linear classification, *Journal of Machine Learning Research*, Volume 9, Pages 1871-1874.
38. Fernandez-Escamilla AM, Rousseau F, and Schymkowitz JSL, 2004, Prediction of sequence-dependent and mutational effects on the aggregation of peptides and proteins, *Nature Biotechnology*, Volume 22, Pages 1302–1306.

39. Fooks HM, Martin AC, Woolfson DN, Sessions RB and Hutchinson EG, 2006, Amino acid pairing preferences in parallel beta-sheets in proteins, *Journal of Molecular Biology*, Volume 356, Issue 1, Pages 32-44.
40. Garg A, Kaur H, and Raghava GP, 2005, Real value prediction of solvent accessibility in proteins using multiple sequence alignment and secondary structure, *Proteins*, Volume 61, Issue 2, Pages 318-324.
41. Gromiha MM and Selvaraj S, 1998, Protein secondary structure prediction in different structural classes, *Protein Engineering*, Volume11, Issue 4, Pages 249-251.
42. Gruber A, Durham AM, Huynh C and Bethesda HAP, 2008, *Bioinformatics in tropical disease research*, NCBI.
43. Goulden CH, 1956, *Methods of Statistical Analysis,* 2nd edition*,* Wiley, Pages 50-55.
44. Gunasekaran K, Nagarajaram HA, Ramakrishnan C, and Balaram P, 1998, Stereochemical punctuation marks in protein structures: glycine and proline containing helix stop signals, *Journal of Molecular Biology*, Volume 275, Pages 917–932.
45. Hall M and Smith L, 1999, Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper, *Proc. FLAIRS*, Pages 235-239.
46. Hall M, 2000, Correlation-based feature selection for discrete and numeric class machine learning, *Proceedings of Seventeenth International Conference on Machine Learning (ICML)*, Pages 359–366.
47. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, and Witten IH, 2009, The WEKA Data Mining software: an update, *SIGKDD Explorations*, Volume 11, Issue 1, Pages 10-18.

48. Ho BK and Curmi PMG, 2002, Twist and Shear in b-Sheets and b-Ribbons, *Journal of Molecular Biology*, Volume 317, Pages 291-308.

49. Holmes G and Nevill-Manning CG, 1995, Feature selection via the discovery of simple classification rules, *Proceedings of International Symposium on Intelligent Data Analysis* (IDA-95).

50. Hu X and Li Q, 2008, Using support vector machine to predict beta- and gamma-turns in proteins, *Journal of Computational Chemistry*, Volume 29, Issue 12, Pages 1867-1875.

51. Hubbard TJ, 1994, Use of β-strand interaction pseudo-potentials in protein structure prediction and modeling, Proceedings of the Biotechnology Computing Track, *Protein Structure Prediction MiniTrack of the 27th HICSS*, IEEE Computer Society Press, Pages 336–354.

52. Hutchinson EG, Sessions RB, Thornton JM and Woolfson DN, 1998, Determinants of strand register in antiparallel beta-sheets of proteins, *Protein Science*, Volume 7, Pages 2287–2300.

53. Ivankov DN and Finkelstein AV, 2004, Prediction of protein folding rates from the amino-acid sequence-predicted secondary structure, *Proceedings of National Academic Sciences*, Volume 101, Pages 8942-8944.

54. Johansson F and Toh H, 2010, a comparative study of conservation andvariation scores, *BMC Bioinformatics* , Volume 11, Article No.388.

55. Jones DT, 1999, Protein secondary structure prediction based on position-specific scoring matrices, *Journal of Molecular Biology*, Volume 292, Issue 2, Pages 195-202.

56. Kabsch W and Sander C, 1983, Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features*, Biopolymers*, Volume 22, Issue 12, Pages 2577-2637.

57. Kamat A and Lesk A, 2007, Contact patterns between helices and strands of sheet define protein folding patterns, *Proteins*, Volume 66, Pages 869-876.

58. Kedarisetti K, Kurgan LA, and Dick S, 2006, A Comment on Prediction of protein structural classes by a new measure of information discrepancy, *Computational Biology and Chemistry*, Volume 30, Issue 5, Pages 393-394.

59. Kedarisetti K, Kurgan L, and Dick S, 2006, Classifier ensembles for protein structural class prediction with varying homology, *Biochemical Biophysical Research Communication*, Volume 348, Issue 3, Pages 981-988.

60. Kedarisetti K, Dick S, and Kurgan LA, 2008, Searching for factors that distinguish disease-prone and disease-resistant prions via sequence analysis, *Bioinformatics and Biology Insights*, Volume 2, Pages 133-144.

61. Kedarisetti KD, Mizianty M, Dick S and Kurgan L, 2011, Improved sequence-based prediction of strand residues, *Journal of Bioinformatics and Computational Biology*, 9, Issue 1, Pages 67-89.

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

63. Klebe G, 2000, Recent developments in structure based drug design, *Journal of Molecular Medicine*, Volume 78, Issue 5, Pages 269-281.

64. Kohavi R and John G, 1996, Wrappers for Feature Subset Selection, *Artificial Intelligence journal,* special Supplement on relevance, Volume 97, Pages 273-324.

65. Kohavi R and Provost F, 1998, Editorial for the special Supplement on applications of *Machine Learning and the knowledge Discovery process*, Glossary of Terms.

66. Koh IY, Eyrich VA, Marti-Renom MA, Przybylski D, Madhusudhan MS, Eswar N, Graña O, Pazos F, Valencia A, Sali A, and Rost B, 2003, EVA: evaluation of protein structure prediction servers*, Nucleic Acids Research*, Volume 31, Pages 3311-3315.

67. Kortemme T, Ramirez-Alvarado M, and Serrano L, 1998, Design of a 20-amino acid, Three-stranded β-sheet protein, *Science*, Volume 281, Pages 253–256.

68. Koonin EV and Galperin MY, 2003, *Sequence - Evolution - Function: Computational Approaches in Comparative Genomics*, published by Kluwer Academic.

69. Kuhlman B, Dantas G, Ireton GC, Varani G, Stoddard BL and Baker D, 2003, Design of a novel globular protein fold with atomic-level accuracy, *Science*, Volume 302, Pages 1364-1368.

70. Kurgan L, Razib AA,Aghakhani S,Dick S,Mizianty M and Jahandideh S, 2009, CRYSTALP2: sequence-based protein crystallization propensity prediction, *BMC Structural Biology,* Volume 9, Article No. 50.

71. La D and Livesay DR, 2005, Predicting functional sites with an automated algorithm suitable for heterogeneous datasets, *BMC Bioinformatics*, Volume 6, Article No. 116.

72. Laskowski RA, Watson JD, and Thornton JM, 2005, ProFunc: a server for predicting protein function from 3D structure, *Nucleic Acids Research,* Volume 33, Pages 89–93.

73. Lengauer T and Zimmer R, 2000, Protein structure prediction methods for drug design, *Briefings in Bioinformatics*, Volume 1, Issue 3, Pages 275-288.

74. Lesk AM., 1997, CASP2: report on ab initio predictions, *Proteins,* Supplement 1, Pages 151-66.

75. Li W, Jaroszewski L and Godzik A, 2002, Tolerating some redundancy significantly speeds up clustering of large protein databases, *Bioinformatics,* Volume 18, Pages 77-82.

76. Lifson S and Sander C, 1979, Antiparallel and parallel beta-strands differ in amino acid residue preferences, *Nature*, Volume 282, Pages 109-111.

77. Lin K, Simossis VA, Taylor WR, and Heringa J, 2005, A Simple and fast secondary structure prediction algorithm using hidden neural networks, *Bioinformatics,* Volume 21, Issue 2, Pages 152-159.

78. Lippi M and Frasconi P, 2009, Prediction of protein beta-residue contacts by markov logic networks with grounding specific weights, *Bioinformatics*, Volume 25, Issue 18, Pages 2326-33.

79. Liu H and Setiono R, 1996, A probabilistic approach to feature selection—A filter solution, *Proceedings of International Conference on Machine Learning*, Pages 319–327.

80. Liu H, Li J and Wong L, 2002, A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns, *Genome Informatics*, Volume 13, Pages 51-60.

81. Liu Y, Carbonell J, Klein-Seetharaman J, and Gopalakrishnan V, 2004, Comparison of probabilistic combination methods for protein secondary structure prediction, *Bioinformatics*, Volume 20, Issue 17, Pages 3099-107.

82. Lodish H, Berk A, Zipursky SL Matsudaira, 2003, *Molecular Cell Biology*, 5th edition, W.H. Freeman & Company.

83. MacCallum R.M, 2004, Striped sheets and protein contact prediction, *Bioinformatics*, Volume 20, Pages i224-i231.

84. Madera M, Calmus R, Thiltgen G, Karplus K and Gough J, 2010, Improving protein secondary structure prediction using a simple k-mer model, *Bioinformatics*, Volume 26, Issue 5, Pages 596-602.

85. Mandel-Gutfreund Y, Zaremba SM, and Gregoret LM, 2001, Contributions of residue pairing to beta-sheet formation: conservation and co-variation of amino acid residue pairs on antiparallel beta-strands, *Journal of Molecular Biology*, Volume 305, Pages 1145-59.

86. Mandel-Gutfreund Y and Gregoret LM, 2002, On the significance of alternating patterns of polar and non-polar residues in beta-strands, *of Molecular Biology,* Volume 323, issue 3, Pages 453–461.

87. Mathews BB, 1975, Comparison of the predicted and observed secondary structure of T4 phage lysozyme, *Biochimica et Biophysica Acta,* Volume 405, Pages 442-451.

88. Max N, Hu C, Kreylos O and Crivelli S, 2010, BuildBeta – A system for automatically constructing beta sheets, *Proteins*, Volume 78, Issue 3, Pages 559-574.

89. McGuffin LJ, Bryson K and Jones DT, 2000, The PSIPRED protein structure prediction server, *Bioinformatics*, Volume 16, Pages 404-405.

90. McGuffin LJ and Jones DT, 2003, Benchmarking secondary structure prediction for fold recognition, *Proteins*, Volume 52, Pages 166-175.

91. Merkel JS, Sturtevant JM and Regan L, 1999, Sidechain interactions in parallel β- sheets: the energetics of cross-strand pairings, *Structural Fold. Description*, Volume 7, Pages 1333–1343.

92. Merkel J.S and Regan L, 2000, Modulating protein folding rates in vivo and in vitro by side-chain interactions between the parallel beta strands of green

fluorescent protein, *Journal of Biological Chemistry,* Volume 275, Pages 29200–29206.

93. Mizianty M and Kurgan L, 2009, Modular prediction of protein structural classes from sequences of twilight-zone identity with predicting sequences, *BMC Bioinformatics*, Volume 10, Article No. 414.
94. Mizianty M, Stach W, Chen K, Kedarisetti KD, Miri Disfani F and Kurgan L, 2010, Improved sequence-based prediction of disordered regions with multilayer fusion of multiple information sources, *Bioinformatics*, Volume 26, Issue 18, Pages i489-i496.
95. Mizianty M and Kurgan L, 2011, Improved identification of outer membrane beta barrel proteins using primary sequence, predicted secondary structure and evolutionary information, *Proteins*, Volume 79, Issue 1, Pages 294–303.
96. Montgomerie S, Sundararaj S, Gallin WJ, and Wishart DS, (2006), Improving the accuracy of protein secondary structure prediction using structural alignment, *BMC Bioinformatics*, Volume 7, Article No. 301.
97. Montgomerie S, Cruz JA, Shrivastava S, Arndt D, Berjanskii M, and Wishart DS, 2008, PROTEUS2: a web server for comprehensive protein structure prediction and structure-based annotation, *Nucleic Acids Research*, Volume36, Pages w202–w209.
98. Moult J, Fidelis K, Kryshtafovych A, Rost B, and Tramontano A, 2009, Critical assessment of methods of protein structure prediction-Round VIII, *Proteins*, Volume 77, Supplement 9, Pages 1-4.
99. Ofer D and Zhou Y, 2007, Achieving 80% ten-fold cross-validated accuracy for secondary structure prediction by large-scale training, *Proteins*, Volume 66, Pages 838-845.

100. Pearson ES and Hartley HO, 1972, *Biometrika Tables for Statisticians*, Volume 2, Page118.

101. Pei J and Grishin NV, 2001, AL2CO: calculation of positional conservation in a protein sequence alignment, *Bioinformatics*, Volume17, Issue 8, Pages 700-712.

102. Peng K, Radivojac P, Vucetic S, Dunker AK, and Obradovic Z, 2006, Length-dependent prediction of protein intrinsic disorder, *BMC Bioinformatics*, Volume 7, Article No. 208.

103. Petsko G and Ringe D, 2004, Protein *Structure and Function: Primers*, New science press ltd.

104. Pintar A, Carugo O and Pongor S, 2003, DPX, for the analysis of the protein core, *Bioinformatics*, Volume 19, Pages 313-314.

105. Pollastri G, Przybylski D, Rost B and Baldi P, 2002, Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles, *Proteins*, Volume 47, Pages 228-235.

106. Pollastri G and McLysaght A, 2005, Porter: a new, accurate server for protein secondary structure prediction, *Bioinformatics*, Volume 21, Pages 1719-1720.

107. Pruitt K, Tatusova T, and Maglott D, 2002, The Reference Sequence (RefSeq) Project, *National Center for Biotechnology Information*.

108. Punta M and Rost B, 2005, PROFcon: novel prediction of long-range contacts, *Bioinformatics*, Volume 21, Issue 13, Pages 2960–2968.

109. Qi Y, Bar-Joseph1 Z, and Klein-Seetharaman J, 200), Evaluation of different biological data and computational classification methods for use in protein interaction prediction, *Proteins: Structure, Function, and Bioinformatics*, Volume 63, Pages 490–500.

110. Rhodes G, 2006, Crystallography Made Crystal Clear, *A Guide for Users of Macromolecular Models*, Third Edition, Elsevier/Academic Press

111. Rost B and Sander C, 1993, Prediction of protein secondary structure at better than 70% accuracy, *Journal Molecular Biology*, Volume 232, Pages 584–599.

112. Rost B and Sander C, 1996, Bridging the protein-sequence–structure gap by structure predictions, *Annual Revision on Biophysics and Biomolecular Structures*, Volume 25, Pages 113–136.

113. Rost B, 1996, PHD: Predicting one-dimensional protein structure by profile based neural networks, *Methods in Enzymology*, Volume 266, Pages 525-539.

114. Rost B and Sander C, 2000, Third generation prediction of secondary structure, *Methods Molecular Biology*, Volume 143, Pages 71-95 .

115. Rost B, 2001, Review: protein secondary structure prediction continues to raise, *Journal of Structural Biology*, Volume 134, Pages 204–218.

116. Rost B and Eyrich VA, 2001, EVA: Large-scale analysis of secondary structure prediction, *Proteins*, Volume 5, Pages 192–199.

117. Rost B, Liu J, Przybylski D, Nair R, Wrzeszczynski KO, Bigelow H, and Ofran Y, 2003, Prediction of protein structure through evolution, *Book chapter in handbook of chemoinformatics from data to knowledge*, Pages 1789–1811.

118. Rost B, 2009, Prediction of protein structure in 1D: secondary structure, membrane regions, and solvent accessibility, *Structural Bioinformatics*, 2nd edition, Gu J, Bourne PE (eds.), Pages 679-714.

119. Rupp B and Gussa JM, 2011, Biomolecular Crystallography: Principles, Practice, and Application to Structural Biology, 2011, online publication, *http://www.amazon.com/Biomolecular-Crystallography-Principles-Application-Structural/dp/0815340818/.*

120. Ruczinski I, Kooperberg C, Bonneau R, and Baker D, 2002, Distributions of beta sheets in proteins with application to structure prediction*, Proteins*, Volume 48, Pages 85–97.

121. Sanchez VD, 2003, Advanced Support Vector Machines and Kernel Methods, *Neurocomputing,* Volume 55, Pages 5–20.

122. Schlessinger A, Punta M, Yachdav G, Kajan L and Rost B, 2009, Improved disorder prediction by combination of orthogonal approaches, *PLoS One*, Volume 4, Issue 2, Article No.e4433.

123. Selbig J, Mevissen T, and Lengauer T, 1999, Decision tree-based formation of consensus protein secondary structure prediction, *Bioinformatics*, Volume 12, Pages 1039-1046.

124. Shapiro SS and Wilk MB, 1965, An analysis of variance test for normality (complete samples), *Biometrika*, Volume 52, Pages 591–611.

125. Shen HB and Chou KC, 2006, Ensemble classifier for protein fold pattern recognition. *Bioinformatics*, Volume 22, Supplement 14, Pages 1717-1722.

126. Shen HB and Chou JJ, 2008, MemBrain: improving the accuracy of predicting transmembrane helices, *PLoS One*, Volume 3, Issue 6, Article No. e2399.

127. Singh R, Xu J, and Berger B, 2006, Struct2net: integrating structure into protein-protein interaction prediction, *Pacific Symposium on Biocomputing*, Pages 403–414.

128. Smith CK and Regan L, 1995, Guidelines for protein design: The energetics of β sheet side chain interactions, *Science*, Volume 270, Pages 980–982.

129. Smith CK and Regan L, 1997, Construction and design of β -sheets, *Accounts of Chemical Research,* Volume 30, Page 153.

130. Stefani M and Dobson CM, 200), Protein aggregation and aggregate toxicity: new insights into protein folding, misfolding diseases and biological evolution, *Journal of Molecular Medicine*, Volume 81, Pages 678–699.

131. Steward RE and Thornton JM, 200), Prediction of strand pairing in antiparallel and parallel beta-sheets using information theory, *Proteins*, Volume 48, Pages 178–191.

132. Stöhr J, Weinmann N, Wille H, Kaimann T, Nagel-Steger L, Birkmann E, Panza G, Prusiner SB, Eigen M, and Riesner D, 2008, Mechanisms of prion protein assembly into amyloid, *Proceedings of National Academic Science*, Volume 105, Issue7, Pages 2409-2414.

133. Stormo GG, Schneider TD, Gold L, and Ehrenfeucht A, 1982, Use of the 'Perceptron' algorithm to distinguish translational initiation sites in E. Coli, *Nucleic Acids Research*, Volume 10, Pages 2997-3012.

134. Tang Z, Li T, Liu R, Xiong W, Sun J, Zhu Y, and Chen G, 2011, Improving the performance of β-turn prediction using predicted shape strings and a two-layer support vector machine model, *BMC Bioinformatics*, Volume 12, Article No. 283.

135. Tegge A N, Wang Z, Eickholt J and Cheng J, 2009, NNcon: improved protein contact map prediction, *Nucleic Acids Research,* Volume 37, Web Server issue w515–w518.

136. Varrazzo D, Bernini A, Spiga O, Ciutti A, Chiellini S, Venditti V, Bracci L and Niccolai N, 2005, Three-dimensional computation of atom depth in complex molecular structures, *Bioinformatics*, Volume 21, Issue 12, Pages 2856-2860.

137. Vullo A, Walsh I, and Pollastri G, 2006, A two-stage approach for improved prediction of residue contact maps, *BMC Bioinformatics*, Volume 7, Article No.180.

138. Vyas J, Gryk MR and MSchiller MR, 2009, *VENN, a tool for titrating sequence conservation onto protein structures,* Nucleic Acids Research, Vol. 37, Article No.18.

139. Ward JJ, McGuffin LJ, Buxton BF and Jones DT, 2003, Secondary structure prediction with support vector machines, *Bioinformatics*, Volume 19, Issue 13, Pages 1650-1655.

140. Wang K and Samudrala R, 2006, Incorporating background frequency improves entropy-based residue conservation measures, *BMC Bioinformatics*, Volume 7, Article No. 385.

141. Wilcoxon F, 1945, Individual comparisons by ranking methods, *Biometric*, Volume 1, Issue 6, Pages 80–83.

142. Wouters MA and Curmi PM, 1995, An analysis of side chain interactions and pair correlations within antiparallel beta-sheets: the differences between backbone hydrogen-bonded and non-hydrogen-bonded residue pairs, *Proteins: Structural Functional Genetics,* Volume 22, Pages. 119–131.

143. Wu ST and Zhang Y, 2008, MUSTER: improving protein sequence profile–profile alignments by using multiple sources of structure information, *Proteins*, Volume 72, Issue 2, Pages 547-556.

144. Xue B, Dunbrack RL, Williams RW, Dunker AK and Uversky VN, 2010, PONDR-FIT: A meta-predictor of intrinsically disordered amino acids, *Biochim Biophys Acta*. Volume 1804, Issue 4, Pages 996-1010.

145. Yuan Z and Wang ZX, 2008, Quantifying the relationship of protein burying depth and sequence, *Proteins*, Volume 70, Pages 509-516.

146. Zaremba SM and Gregoret LM, 1999, Context-dependence of amino acid residue pairing in antiparallel β-sheets, *Journal of Molecular Biology*, Volume 291, Pages 463–479.

147. Zemla A, Venclovas C, Fidelis K, and Rost B, 1999, A modified definition of Sov, a segment-based measure for protein secondary structure prediction assessment, *Proteins*, Volume 34, Issue 2, Pages 220-223.

148. Zhang C and Kim S, 2000, The anatomy of protein beta-sheet topology, *Journal of Molecular Biology,* Volume 2, Pages 1075–1089.

149. Zhang H, Zhang T, Chen K, Shen S, Ruan J and Kurgan L, 2008, Sequence based residue depth prediction using evolutionary information and predicted secondary structure, *BMC Bioinformatics,* Volume 9, Article No. 388.

150. Zhang H, Zhang T, Chen K, Kedarisetti KD, Mizianty MJ, Bao Q, Stach W, Kurgan L, 2011, Critical assessment of high-throughput standalone methods for secondary structure prediction, *Briefings in Bioinformactics,* Volume 12, Issue 6, Pages 672-688.

151. Zhang N, Ruan J, Wu J and Zhang T, 2007, Sheetspair: A Database Of Amino Acid Pairs In Protein Sheet Structures, *Data Science Journal*, Volume 6, Issue 15, Pages s589-s595.

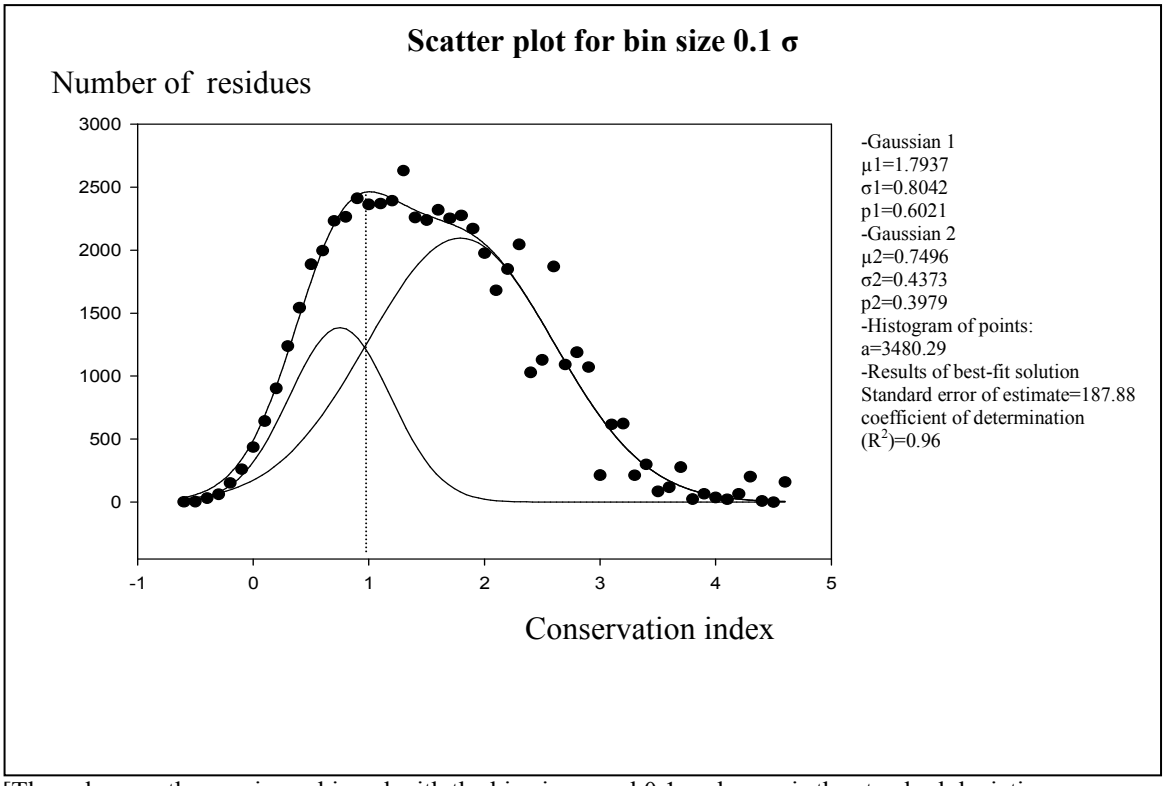
152. Zhang N, Ruan J, Duan G, Gao S and Zhang T, 2009,The interstrand amino acid pairs play a significant role in determining the parallel or antiparallel orientation of β-strands, *Biochemical and Biophysical Research Communications*, Volume 386, Pages 537–543.

153. Zhang N, Duan G, Gao S, Ruan J and Zhang T, 2010, Prediction of the parallel/antiparallel orientation of beta-strands using amino acid pairing preferences and support vector machines, *Journal of Theoretical Biology*, Volume 263, Issue 3, Pages 360-368.

154. Zhang Y, 2009, I-TASSER: Fully automated protein structure prediction in CASP8, *Proteins*, Volume 77, Supplement 9, Pages 100–113.

155. Zheng C and Kurgan L, 2008, Prediction of ß-turns at over 80% accuracy based on an ensemble of predicted secondary structures and multiple alignments. BMC *Bioinformatics*, Volume 9, Article No. 430.

Appendix



[The values on the x-axis are binned with the bin size equal 0.1σ where σ is the standard deviation. The corresponding number of residues is shown on the y-axis. Based on (Jimin et al., 2001) and using the Sigmaplot software, these data were fitted into the sum of two Gaussian distributions. (g1 + g2): f=a*(p1*exp(-.5*((x-μ1)/ σ1)2)+p2*exp(-.5*((x- μ2)/ σ2)2)), where μ1 and μ1 are means, σ1 and σ1 are standard deviations and p1 and p2 are coefficients in the sum of two Gaussians. These two Gaussian distributions serve as an approximation of the low conservation and the high conservation components, respectively, and a is parameter that describes bin size*number of residues. The parameters of the best fit are shown on the right side. The dashed line shows the threshold that is used to binarize the conservation scores.]

Supplementary Figure 0-1 The relative entropy-based conservation score values in a histogram, which are shown using black dots.

ANOVA results for bin size 0.1σ:

Nonlinear Regression - Dynamic Fitting **Tuesday, July 06, 2010, 5:28:09 PM**

Data Source: Data 1 in Notebook6
Equation: User-Defined, Weighted Sum 2 Gaussian
f=a*(p1*exp(-.5*((x-x10)/b1)^2)+p2*exp(-.5*((x-x20)/b2)^2))

Dynamic Fit Options:

Total Number of Fits	200
Maximum Number of Iterations	200

Parameter Ranges for Initial Estimates:

	Minimum	Maximum
a	-2631.0000	7893.0000
b1	0.0000	3.4091
b2	0.0000	3.4091
x10	-1.3000	3.9000
x20	-1.3000	3.9000
p1	0.0000	1.5000
p2	0.0000	1.5000

Summary of Fit Results:

Converged	98.5%
Singular Solutions	83.0%
Ill-Conditioned Solutions	15.5%
Iterations Exceeding 200	1.5%

Results for the Overall Best-Fit Solution:

R	Rsqr	Adj Rsqr	Standard Error of Estimate
0.9823	0.9649	0.9603	187.8832

	Coefficient	Std. Error	t	P
a	3480.2911	1444.5066	2.4093	0.0200
b1	0.8042	0.1150	6.9950	<0.0001
b2	0.4373	0.0914	4.7857	<0.0001
x10	1.7937	0.2069	8.6715	<0.0001
x20	0.7496	0.0629	11.9177	<0.0001
p1	0.6021	0.3040	1.9807	0.0536
p2	0.3979	2.0012E-006	198825.6620	<0.0001

Analysis of Variance:

Analysis of Variance:

	DF	SS	MS
Regression	7	106618413.3380	15231201.9054
Residual	46	1623803.6620	35300.0796
Total	53	108242217.0000	2042305.9811

Corrected for the mean of the observations:

	DF	SS	MS	F	P
Regression	6	44663058.4512	7443843.0752	210.8733	<0.0001
Residual	46	1623803.6620	35300.0796		
Total	52	46286862.1132	890131.9637		

Statistical Tests:

Normality Test (Shapiro-Wilk) Failed (P = 0.0004)

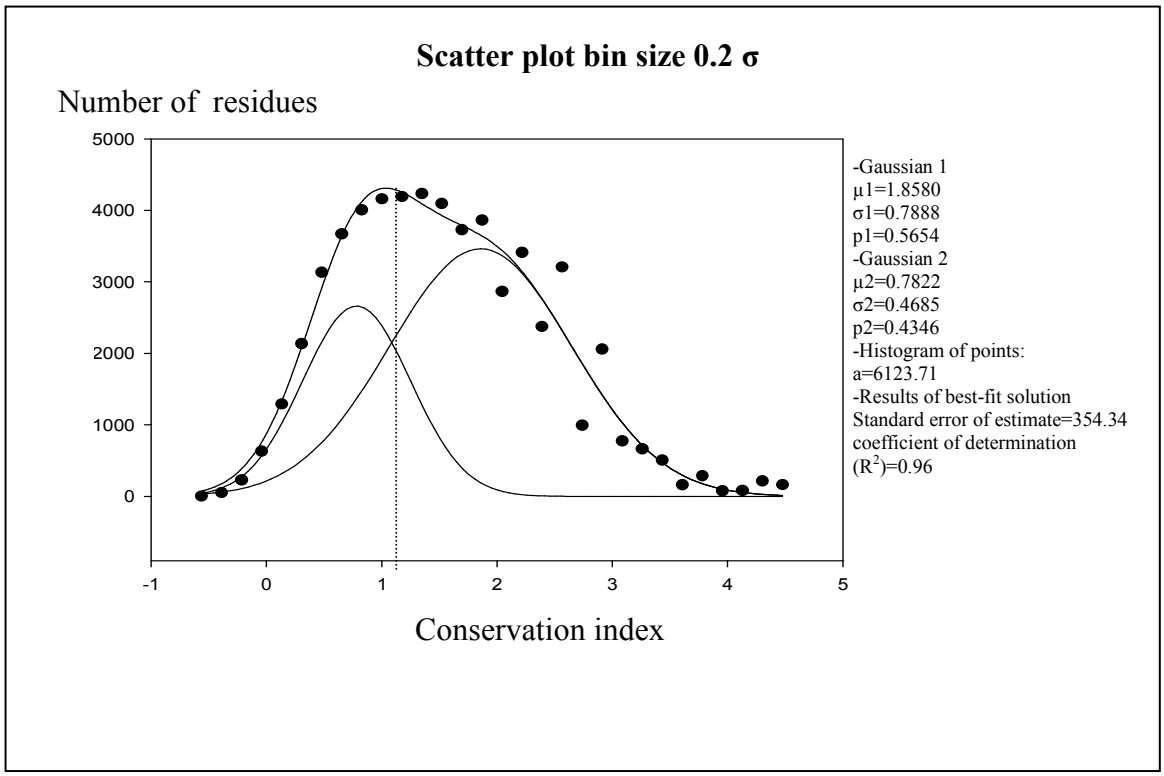
W Statistic= 0.9019 Significance Level = 0.0500

Constant Variance Test Passed (P = 0.3232)

Fit Equation Description:

```
[Variables]
x = col(1)
y = col(2)
reciprocal_y = 1/abs(y)
reciprocal_ysquare = 1/y^2
'Automatic Initial Parameter Estimate Functions
peaksign(q)=if(total(q)>q[1], 1, -1)
xatymin(q,r)=xatymax(q,max(r)-r)
[Parameters]
a = if(peaksign(y)>0, max(y), min(y)) "Auto {{previous: 3480.29}}
b1 = fwhm(x,abs(y))/2.2 "Auto {{previous: 0.80419}}
b2 = fwhm(x,abs(y))/2.2 "Auto {{previous: 0.437316}}
x10 = if(peaksign(y)>0, xatymax(x,y), xatymin(x,y)) "Auto {{previous: 1.7937}}
x20 = if(peaksign(y)>0, xatymax(x,y), xatymin(x,y)) "Auto {{previous: 0.749613}}
p1 = 0.5 ' {{previous: 0.602113}}
p2 = 0.5 ' {{previous: 0.397887}}
[Equation]
f=a*(p1*exp(-.5*((x-x10)/b1)^2)+p2*exp(-.5*((x-x20)/b2)^2))
fit f to y
"fit f to y with weight reciprocal_y
"fit f to y with weight reciprocal_ysquare
[Constraints]
b1>0
b2>0
p1>0
p2>0
p1+p2=1
[Options]
tolerance=1e-010
stepsize=1
iterations=200
```

Number of Iterations Performed = 36



[The values on the x-axis are binned with the bin size equal 0.2σ where σ is the standard deviation. The corresponding number of residues is shown on the y-axis. Based on (Jimin et al., 2001) and using the Sigmaplot software, these data were fitted into the sum of two Gaussian distributions. (g1 + g2): f=a*(p1*exp(-.5*((x-μ1)/ σ1)2)+p2*exp(-.5*((x- μ2)/ σ2)2)), where μ1 and μ1 are means, σ1 and σ1 are standard deviations and p1 and p2 are coefficients in the sum of two Gaussians. These two Gaussian distributions serve as an approximation of the low conservation and the high conservation components, respectively, and a is parameter that describes bin size*number of residues. The parameters of the best fit are shown on the right side. The dashed line shows the threshold that is used to binarize the conservation scores]

Supplementary Figure 0-2 The relative entropy-based conservation score values in a histogram, which are shown using black dots.

ANOVA results for bin size 0.2σ:

Nonlinear Regression - Dynamic Fitting Tuesday, July 06, 2010, 5:21:06 PM

Data Source: Data 1 in Notebook5
Equation: User-Defined, Weighted Sum 2 Gaussian
f=a*(p1*exp(-.5*((x-x10)/b1)^2)+p2*exp(-.5*((x-x20)/b2)^2))

Dynamic Fit Options:

Total Number of Fits	200
Maximum Number of Iterations	200

Parameter Ranges for Initial Estimates:

	Minimum	Maximum
a	-4236.0000	12708.0000
b1	0.0000	3.5542
b2	0.0000	3.5542
x10	-1.3482	4.0447
x20	-1.3482	4.0447
p1	0.0000	1.5000
p2	0.0000	1.5000

Summary of Fit Results:

Converged	99.0%
Singular Solutions	87.5%
Ill-Conditioned Solutions	11.5%
Iterations Exceeding 200	0.5%
Inner-Loop Failures	0.5%

Results for the Overall Best-Fit Solution:

R	Rsqr	Adj Rsqr	Standard Error of Estimate
0.9811	0.9626	0.9528	355.3416

	Coefficient	Std. Error	t	P
a	6123.7125	3672.0365	1.6677	0.1089
b1	0.7888	0.1882	4.1922	0.0003
b2	0.4685	0.1224	3.8294	0.0009
x10	1.8580	0.3454	5.3787	<0.0001
x20	0.7822	0.1178	6.6384	<0.0001
p1	0.5654	0.4308	1.3123	0.2024
p2	0.4346	1.9827E-006	219199.8522	<0.0001

Analysis of Variance:

Analysis of Variance:

	DF	SS	MS
Regression	7	184154581.7563	26307797.3938
Residual	23	2904155.2437	126267.6193
Total	30	187058737.0000	6235291.2333

Corrected for the mean of the observations:

	DF	SS	MS	F	P
Regression	6	74692480.9230	12448746.8205	98.5902	<0.0001
Residual	23	2904155.2437	126267.6193		
Total	29	77596636.1667	2675746.0747		

Statistical Tests:

Normality Test (Shapiro-Wilk) Failed (P = 0.0198)

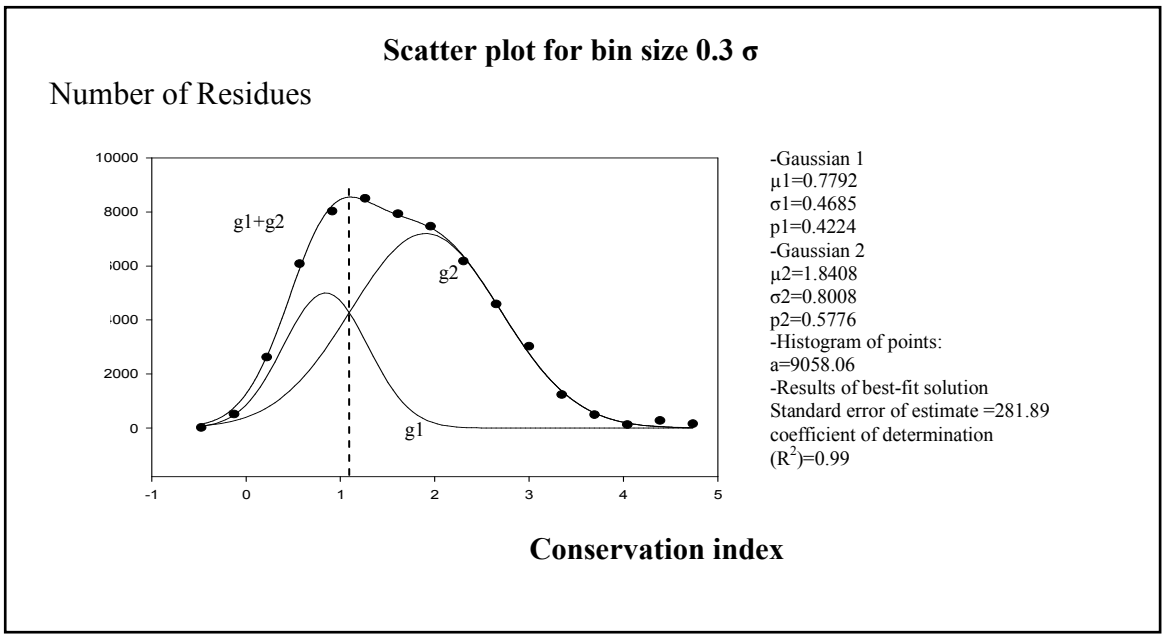
W Statistic= 0.9149 Significance Level = 0.0500

Constant Variance Test Passed (P = 0.2531)

Fit Equation Description:

```
[Variables]
x = col(1)
y = col(2)
reciprocal_y = 1/abs(y)
reciprocal_ysquare = 1/y^2
'Automatic Initial Parameter Estimate Functions
peaksign(q)=if(total(q)>q[1], 1, -1)
xatymin(q,r)=xatymax(q,max(r)-r)
[Parameters]
a = if(peaksign(y)>0, max(y), min(y)) "Auto {{previous: 6123.71}}
b1 = fwhm(x,abs(y))/2.2 "Auto {{previous: 0.788793}}
b2 = fwhm(x,abs(y))/2.2 "Auto {{previous: 0.468531}}
x10 = if(peaksign(y)>0, xatymax(x,y), xatymin(x,y)) "Auto {{previous: 1.85804}}
x20 = if(peaksign(y)>0, xatymax(x,y), xatymin(x,y)) "Auto {{previous: 0.78221}}
p1 = 0.5 ' {{previous: 0.565396}}
p2 = 0.5 ' {{previous: 0.434604}}
[Equation]
f=a*(p1*exp(-.5*((x-x10)/b1)^2)+p2*exp(-.5*((x-x20)/b2)^2))
fit f to y
"fit f to y with weight reciprocal_y
"fit f to y with weight reciprocal_ysquare
[Constraints]
b1>0
b2>0
p1>0
p2>0
p1+p2=1
[Options]
tolerance=1e-010
stepsize=1
iterations=200
```

Number of Iterations Performed = 32



[The values on the x-axis are binned with the bin size equal 0.1σ(a), 0.2σ(b), 0.3σ(c), where σ is the standard deviation. The corresponding number of residues is shown on the y-axis. Based on (Jimin et al., 2001) and using the Sigmaplot software, these data were fitted into the sum of two Gaussian distributions. (g1 + g2): f=a*(p1*exp(-.5*((x-μ1)/ σ1)2)+p2*exp(-.5*((x- μ2)/ σ2)2)), where μ1 and μ1 are means, σ1 and σ1 are standard deviations and p1 and p2 are coefficients in the sum of two Gaussians. These two Gaussian distributions serve as an approximation of the low conservation and the high conservation components, respectively, and a is parameter that describes bin size*number of residues. The parameters of the best fit are shown on the right side. The dashed line shows the threshold that is used to binarize the conservation scores.]

Supplementary Figure 0-3 The relative entropy-based conservation score values in a histogram, which are shown using black dots.

ANOVA results for bin size 0.3σ:

Nonlinear Regression - Dynamic Fitting **Tuesday, July 06, 2010, 12:44:35 PM**

Data Source: Data 1 in Notebook2
Equation: User-Defined, Weighted Sum 2 Gaussian
f=a*(p1*exp(-.5*((x-x10)/b1)^2)+p2*exp(-.5*((x-x20)/b2)^2))

Dynamic Fit Options:

Total Number of Fits	200
Maximum Number of Iterations	200

Parameter Ranges for Initial Estimates:

	Minimum	Maximum
a	-6380.0000	19140.0000

b1	0.0000	2.8433
b2	0.0000	2.8433
x10	-1.3048	3.9144
x20	-1.3048	3.9144
p1	0.0000	1.5000
p2	0.0000	1.5000

Summary of Fit Results:

Converged	97.0%
Singular Solutions	82.0%
Ill-Conditioned Solutions	15.0%
Iterations Exceeding 200	2.0%
Inner-Loop Failures	1.0%

Results for the Overall Best-Fit Solution:

R	**Rsqr**	**Adj Rsqr**	**Standard Error of Estimate**
0.9954	0.9909	0.9867	281.8909

	Coefficient	**Std. Error**	**t**	**P**
a	9058.056318	10849	8.4301E-007	1.0000
b1	0.4685	0.0875	5.3565	0.0001
b2	0.8008	0.1261	6.3490	<0.0001
x10	0.7792	0.0780	9.9924	<0.0001
x20	1.8408	0.2370	7.7668	<0.0001
p1	0.4224	844534.3121	5.0021E-007	1.0000
p2	0.577611	54625.0448	5.0021E-007	1.0000

Analysis of Variance:

Analysis of Variance:

	DF	**SS**	**MS**
Regression	7	276590790.9124	39512970.1303
Residual	13	1033012.0876	79462.4683
Total	20	277623803.0000	13881190.1500

Corrected for the mean of the observations:

	DF	**SS**	**MS**	**F**	**P**
Regression	6	112397639.6624	18732939.9437	235.7458	<0.0001
Residual	13	1033012.0876	79462.4683		
Total	19	113430651.7500	5970034.3026		

Statistical Tests:

Normality Test (Shapiro-Wilk) Passed (P = 0.7210)

W Statistic= 0.9684 Significance Level = 0.0500

Constant Variance Test Passed (P = 0.2804)

Fit Equation Description:
[Variables]

```
x = col(1)
y = col(2)
reciprocal_y = 1/abs(y)
reciprocal_ysquare = 1/y^2
'Automatic Initial Parameter Estimate Functions
peaksign(q)=if(total(q)>q[1], 1, -1)
xatymin(q,r)=xatymax(q,max(r)-r)
[Parameters]
a = if(peaksign(y)>0, max(y), min(y)) "Auto {{previous: 9058.06}}
b1 = fwhm(x,abs(y))/2.2 "Auto {{previous: 0.468465}}
b2 = fwhm(x,abs(y))/2.2 "Auto {{previous: 0.80078}}
x10 = if(peaksign(y)>0, xatymax(x,y), xatymin(x,y)) "Auto {{previous: 0.779216}}
x20 = if(peaksign(y)>0, xatymax(x,y), xatymin(x,y)) "Auto {{previous: 1.84077}}
p1 = 0.5 ' {{previous: 0.422445}}
p2 = 0.5 ' {{previous: 0.577555}}
[Equation]
f=a*(p1*exp(-.5*((x-x10)/b1)^2)+p2*exp(-.5*((x-x20)/b2)^2))
fit f to y
"fit f to y with weight reciprocal_y
"fit f to y with weight reciprocal_ysquare
[Constraints]
b1>0
b2>0
p1>0
p2>0
p1+p2=1
[Options]
tolerance=1e-010
stepsize=1
iterations=200

Number of Iterations Performed = 19
```