Protein Secondary Structure Prediction Evaluation and a Novel Transition Site Model with New Encoding Schemes

by

Masood Zamani

A Thesis

presented to

The University of Guelph

In partial fulfilment of requirements

for the degree of

Doctor of Philosophy

in

Computer Science

Guelph, Ontario, Canada

ABSTRACT

Protein Secondary Structure Prediction Evaluation and a Novel Transition Site Model with New Encoding Schemes

Masood Zamani
University of Guelph, 2017

Advisor:
Dr. Stefan C. Kremer

Rapid progress in genomics has led to the discovery of millions of protein sequences while less than 0.2% of the sequenced proteins' structures have been resolved by X-ray crystallography or NMR spectroscopy which are complex, time consuming, and expensive. Employing advanced computational techniques for protein structure prediction at secondary and tertiary levels provides alternative ways to accelerate the prediction process and overcome the extremely low percentage of protein structures that have been determined. State-of the art protein secondary structure (PSS) prediction methods employ machine learning (ML) techniques, compared to early approaches based on statistical information and sequence homology. In this research, we develop a two-stage PSS prediction model based on Artificial Neural Networks (ANNs) and Genetic Programming (GP) through a novel framework of PSS transition sites, and new amino acid encoding schemes derived from the genetic Codon mappings, Clustering and Information theory. PSS transition sites represent structural information of protein backbones, and reduce the input space and learning parameters in the PSS prediction model. PSS transition sites can be utilized in Homology Modeling

(HM) to define the boundary of secondary structure elements. The prediction performance of the proposed method is evaluated by using Q3 and segment overlap (SOV) scores on two standard datasets, RS126 and CB513, and the latest protein dataset, PISCES, compiled with very strict homology measures by which each sequence pair has a similarity below the twilight zone or less than 25%. The experimental results and statistical analyses of the proposed PSS model indicate statistically significant improvements in PSS prediction accuracy compared to the state-of-the-art ML techniques which commonly employ cascaded ANNs and SVMs. The proposed encoding schemes show advantages in extracting sequence and profile information, reducing input parameters and training performances. A successful PSS prediction model can be utilized in homology detection tools for distant protein sequences and protein tertiary structure prediction methods to reduce the complexity of the protein structure prediction which has important applications in medicine, agriculture and the biological sciences.

Acknowledgements

I would like to express my gratitude to my research advisor Dr. Stefan C. Kremer who was abundantly helpful and offered invaluable assistance, support and guidance. I would also like to convey thanks to Dr. David Chiu, Dr. Medhat Moussa and Dr. Lourdes Peña-Castillo, the members of the advisory committee and external examiner, and I gratefully acknowledge their guidance and contribution.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview of Proteins

Proteins are the most essential building blocks of all living organisms. A protein is the endproduct of a biological process by which a fragment of a DNA sequence is used as a template to produce a linear macromolecule chain of amino acids during transcription and translation phases. In living cells, DNA contains the blueprint of genetic information which is transcribed to messenger RNA (mRNA) that contains the program for synthesizing a protein. mRNAs are converted to linear amino acid chains during a biological process called *translation* (Crick et al., 1970). In the next stage, the linear amino acid chains fold into unique tertiary structures known as *native* conformations. The native conformation of a protein is a three-dimensional structure that the protein adopts at the last stage of a folding process, and the final conformation has minimum potential energy, not always, but normally.

During a biological process known as folding pathways, the linear amino acid chain of a protein adopts various intermediate conformations until all atomic forces are stabilized and forming the native conformation. The functional mechanism and

interactions of proteins that determine a number of biological phenomena are precisely related to their structures (Blundell, Bedarkar, Rinderknecht, & Humbel, 1978). The complexities of protein structures are decisive to their vital roles in all living beings. Proteins facilitate chemical reactions, regulate cellular processes and carry signals to and from cells. Also, proteins act as antibodies, small molecules, structural elements that bind cells, and motor elements which help movements.

The tertiary structure of a protein is determined by using the atomic coordinate file of the protein. The data file defines three-dimensional (3D) molecular structures by specifying the positions of each atom in space, typically with X, Y and Z Cartesian coordinates. Also, if two or more protein chains interact, a quaternary structure is formed which is a more complex protein structure.

1.2 Computational Methods for Protein Structures

Rapid and staggering advances in genomics have resulted in the identification of several millions of sequenced proteins in recent years (Bairoch, Apweiler, Wu, Barker, Boeckmann, Ferro, Gasteiger, Huang, Lopez, Magrane et al., 2005). In contrast less than 0.2% of the sequenced proteins' structures have been resolved by experimental methods which are the most accurate techniques at the present time. The ratio of the number of sequenced proteins to the number of proteins with known structures during the recent decade is shown in Figure 1.1. We obtained the data from National Center for Biotechnology Information (NCBI) and Protein Data Bank (PDB) (Sussman et al., 1998).

The experimental methods of protein structure determination such as X-ray crystallography (Bragg, Phillips, Lipson et al., 1975; Blundell & Johnson, 1976) and NMR spectroscopy (Wuthrich, 1986; Baldwin, Weber, St Charles, Xuan, Appella,

Yamada, Matsushima, Edwards, Clore, & Gronenborn, 1991) require complex procedures which are time-consuming and costly (Metfessel & Saurugger, 1993; Johnson, Srinivasan, Sowdhamini, & Blundell, 1994). The application of computational meth-

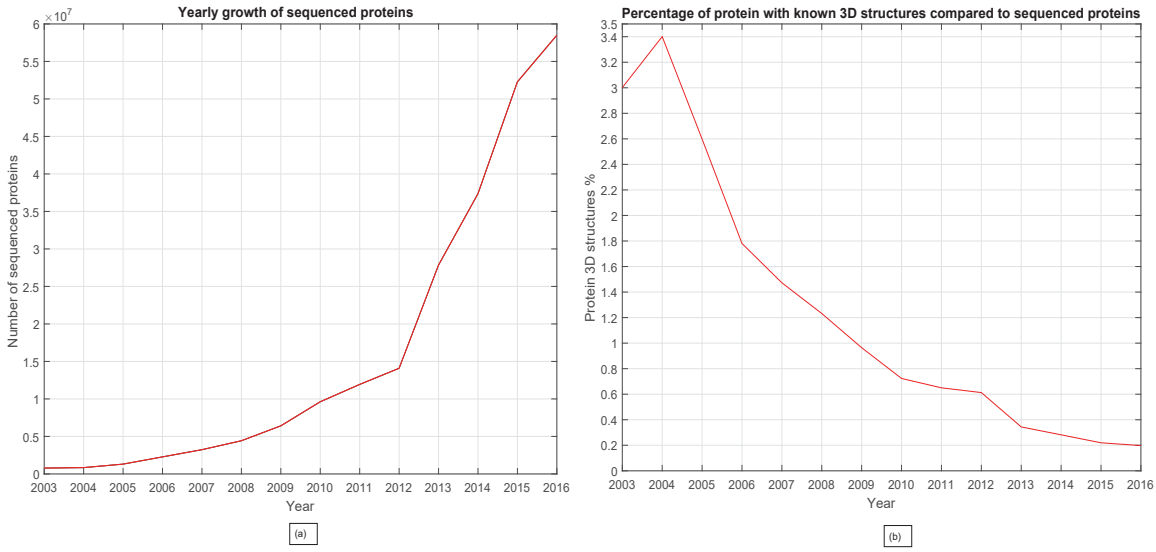

Figure 1.1: The growth ratio – (a) sequenced proteins, (b) protein tertiary structures.

ods in protein structure prediction can be an alternative approach to reduce the gap between the number of sequenced proteins and the number of protein structures ($60m$ vs. $116k$). Unlike experimental techniques, computational methods are capable of providing fast and inexpensive approaches for protein structure prediction with regards to the progress that has been made in computational fields in recent years. Meanwhile, at the present time the outcomes of protein structure prediction by computational techniques using amino acid sequences are limited to short and medium lengths of protein sequences and the accuracies of the predicted protein structures should be validated by the results of the existing experimental techniques of protein structure determination. Moreover, as shown in Figure 1.2 the total number of unique protein folds are declining and a handful of unique protein folds have been

only identified in a few recent years based on CATH and SCOP databases which contain classified protein folds. The trend of new fold identifications indicates potential advantages to employ computational intelligence techniques for protein structure prediction due to the decrease in the number of new patterns that should be learned by the computational intelligence methods.

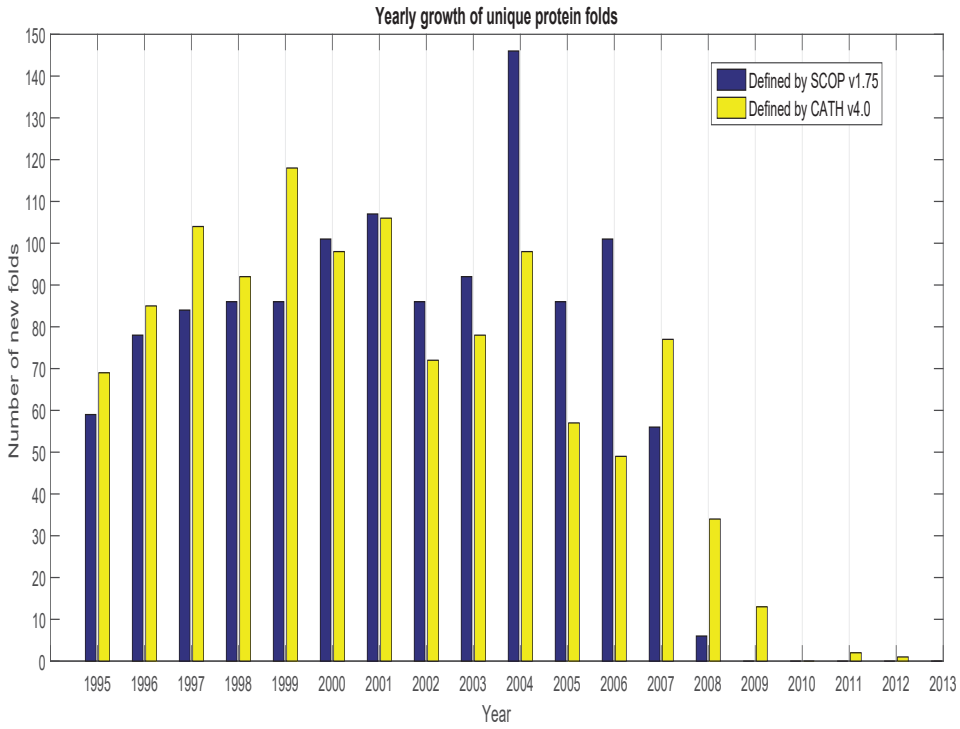

Figure 1.2: The growth of unique protein folds defined in SCOP and CATH databases.

It was speculated based on the experimental evidence that the conformation of a protein can be uniquely determined to a great extent by only the sequence of amino acids (Anfinsen et al., 1973). The hypothesis has been the foundation and motivation to develop various techniques of protein structure prediction for several decades. Protein structure determination from the sequence of amino acids helps greatly to understand the structure-function relationship. Therefore, by changing the structures

of proteins, functions can be added or removed from the proteins, or new proteins can be synthesized to achieve desired functions. Knowing the tertiary structures and functions of proteins has a number of important applications in medicine such as designing drugs and enzymes, agriculture and biological sciences (Chou, 2004; Noble, Endicott, & Johnson, 2004).

1.3 Importance of Protein Secondary Structure

Although gradual progress has been made in protein 3D structure prediction, the outcomes generally have not reached the adequate level of structural accuracy compared to those of the experimental methods of protein structure determination which are the bases of performance evaluation for the quality of protein tertiary structure prediction (Moult, Fidelis, Kryshtafovych, Schwede, & Tramontano, 2016). In general, most tertiary structure prediction methods rely on template structures with a high degree of sequence similarity to a query sequence.

At a structural level, the tertiary structure of a protein can be defined with less complexity by secondary structures which are formed by the hydrogen bonds between the amine hydrogen and carbonyl oxygen atoms of a protein's backbone peptide bonds. Protein secondary structures are consecutive segments of a protein sequence and represent the structural patterns of protein tertiary structures, and the secondary structures are grouped broadly to helices and strands which exhibit distinct geometrical features in form of regular and repetitive folds, and coils which are irregular regions (Pauling, Corey, & Branson, 1951; Pauling & Corey, 1951b).

It was investigated that secondary structures are formed at the early stage of a protein's folding process, and the early formation of secondary structure elements direct the initial structural framework of the protein's 3D structure (Goldenberg, Frieden,

Haack, & Morrison, 1989). The implication of an initial structural framework is the main motivation that protein secondary structure (PSS) prediction methods are utilized in protein tertiary structure modeling techniques to achieve two goals: (1) reducing the computational cost and complexity of 3D structure modeling, and (2) having benefits from the fact that protein structures are more preserved than amino acid sequences. Although homologous proteins have usually higher sequence similarity than non-homologous proteins and consequently structural similarity compared to non-homologous proteins, there are many non-homologous proteins that have structural similarities due to the preservation of their essential functions according to evolutionary divergences.

In *de novo* methods, where protein 3D structures are modeled from amino acid sequences, it has been observed that incorporating the information of accurately predicted secondary structures improves greatly the overall performance (Skolnick, Kolinski, & Ortiz, 1997; Wang, Kudryashev, Li, Egelman, Basler, Cheng, Baker, & DiMaio, 2015). Also, protein 3D structure determination techniques based on protein threading and comparative modeling most often incorporate the information of secondary structure elements (Fischer & Eisenberg, 1996; Koretke, Russell, Copley, & Lupas, 1999; Zhou & Zhou, 2004; Sułkowska, Morcos, Weigt, Hwa, & Onuchic, 2012; Biasini, Bienert, Waterhouse, Arnold, Studer, Schmidt, Kiefer, Cassarino, Bertoni, Bordoli et al., 2014). For example, in protein threading, which is an inverse protein folding, the goal is to find a template structure that is compatible with a given query sequence (Jones, 1999b). Initially, a database of protein structures is searched for protein folds whose pseudo-energies are computed to detect the closest match for the query sequence whose predicted secondary structures are used in the threading procedure.

Moreover, protein secondary structure prediction is often an important compo-

nent of a number of computational biology tools for distant homology detections and multi-sequence alignments (Ginalski, Pas, Wyrwicz, Von Grotthuss, Bujnicki, & Rychlewski, 2003; Simossis & Heringa, 2005; Hargbo & Elofsson, 1999; Zhou & Zhou, 2005).

1.4 State of Secondary Structure Prediction

The accuracy of protein secondary structure (PSS) prediction is commonly measured based on a three-state score (Q_3) by which the eight types of secondary structures, having partially structural similarities, are grouped to helices (H), strands (E) and coils (C). The Q_3 scores is computed by the ratio of the number of residues whose secondary structures are correctly predicted to the total number of residues in an amino acid sequence. Segment overlap (SOV) is another accuracy score to measure the quality of prediction at a segment level, and it results in scores that are often a few percentages lower than Q_3 scores (Zemla, Venclovas, Fidelis, & Rost, 1999). The detailed explanation of PSS evaluation is provided in Chapter 4.

The primary PSS prediction methods were based on two approaches: (1) physico- and stereo-chemical analyses (Lim, 1974a; Ptitsyn & Finkelstein, 1983; Tanaka & Scheraga, 1976), and (2) statistics (Nagano, 1973; Chou & Fasman, 1974a; Garnier, Osguthorpe, Robson et al., 1978; Kabsch, Sander et al., 1983; Gibrat, Garnier, & Robson, 1987; Garnier, Gibrat, Robson et al., 1996; Kloczkowski, Ting, Jernigan, & Garnier, 2002).

By the PSS methods based on physico- and stereo-chemical approaches, a set of extensive rules were derived by using the frequencies of helices and strands observed in a limited number of globular proteins. The prediction rules were based on the principles that govern secondary structure formations with regards to local or long-

range residue interactions. These methods achieved for some time the highest three-state prediction accuracy (Q_3), reaching to 56%, until more protein 3D structures were resolved by experimental methods.

In statistical approaches, the PSS prediction methods derive statistical information from protein sequences and formulate the correlation between sequence composition and the three secondary structures by using Bayesian statistics and Information theory. In the early methods, the likelihoods of residue pairs in specific secondary structures were investigated to quantify residue interactions within short ranges. Then, the residue interactions were combined to compute interacting residue propensities of secondary structure elements. The prediction accuracy of statistical PSS methods ranged from 55% to slightly above 60% based on the Q_3 score and with limited number of globular proteins with known 3D structures, approximately from 20 to 40 protein sequences.

The next class of PSS prediction methods was based on sequence homology when more protein structures resolved by experimental techniques were available. In homologous PSS prediction methods, a protein database is scanned to detect the homology between the segments of a query sequence and those of the template sequences. These methods are based on the hypothesis that short segments of protein sequences with high sequence similarity can have possibly similar structural folds and consequently similar secondary structures even if the compared protein sequences as a whole have low pairwise sequence similarity (Levin, Robson, & Garnier, 1986; Nishikawa & Ooi, 1986). It is based on the fact that structural folds have critical roles in proteins' functions and during evolutionary divergences the essential functions of proteins are likely preserved.

Therefore, in homologous PSS prediction methods, short segments of a query sequences are searched to identify a set of sequence segments with the highest similarity

match to the template segments by using a substitution matrix, and the prediction is made by a nearest-neighbor technique. In a protein database, all sequences are scanned by a fixed-length sliding window which moves one residue at a time along sequences. The homology-based PSS methods achieved an average of 60% prediction accuracy on a limited number of protein sequences.

Later on, a new wave of PSS prediction methods emerged based on various Machine Learning (ML) techniques such as artificial neural networks, support vector machines and hidden markov models (Qian & Sejnowski, 1988; Asai, Hayamizu, & Handa, 1993; Hua & Sun, 2001). ML-based PSS prediction methods that incorporated sequence information improved prediction accuracy to an average of 65% which is higher than those of the previous techniques. However, a significant performance improvement occurred when ML-based methods incorporated protein evolutionary information which increased the prediction accuracy slightly to above 70%.

The prediction accuracy of PSS methods improved further with cascaded architectures, hybrid schemes and consensus approaches which are the state-of-the-art PSS prediction methods. These modification and alternative techniques resulted in a prediction accuracy ranging from 70% to slightly above 80% (Rost, Sander et al., 1993; Rost, 1996; Przybylski & Rost, 2002; Cuff, Barton et al., 2000; Jones, 1999a; Kim & Park, 2003; Pollastri & Mclysaght, 2005; Martin, Gibrat, & Rodolphe, 2006; Montgomerie, Sundararaj, Gallin, & Wishart, 2006; Faraggi, Zhang, Yang, Kurgan, & Zhou, 2012).

However, there are a number of criteria that affect the overall performance of PSS prediction methods which need to be addressed in order to compare their performances. The conditions that should be considered for unbiased comparisons of PSS prediction methods are homology, the architectures of the methods, evaluation schemes, the types of input information, dataset sizes and generalization techniques

which are discussed in detail in Chapter 2.

1.5 Protein Datasets and Evaluation

The protein Data Bank (PDB) is the single worldwide repository and archive of large biological molecules' 3D structures, including proteins and nucleic acids in various formats and detail (Sussman et al., 1998). Initially, PDB contained only seven protein chains with known tertiary structures when it was established in 1971, and the number of the protein structures were increased only to a few dozens during the 1970s.

All protein structure prediction methods, including PSS prediction techniques, rely directly or indirectly on the protein sequences whose structures are known by experimental methods of protein structure determination and deposited in the PDB databases. The early PSS prediction methods which were mostly based on statistical techniques, and the physiochemical and spatial properties of amino acids and secondary structure formations were modeled with a limited number of protein sequences, from 10 to 80 protein chains, whose tertiary structures were known, and the methods' performances were consequently validated insufficiently to some degrees due to the small numbers of available protein structures (Lim, 1974a; Kabsch et al., 1983; Gibrat et al., 1987).

Thereafter, the gradual increase of protein sequences with known structures, although extremely low at the present time compared to available protein sequences, has provided a chance for protein structure prediction methods to take the advantage of the available structural information. However, the relations between proteins' amino acid sequences and structures are very complicated based on the level of protein sequences' pairwise similarities. On one hand, studies show proteins with similar amino acid sequences fold into similar tertiary structures (Zuckerkandl &

Pauling, 1965; Doolittle, 1981, 1986; Chothial & Lesk, 1986), and most protein sequence pairs with more than thirty out of one hundred identical residues were found to have structural similarities (Sander & Schneider, 1991). On the other hand, structure alignments have revealed that there are homologous protein pairs with less than 10% pairwise sequence identity (Holm & Sander, 1996; Hubbard, Murzin, Brenner, & Chothia, 1997). More precisely, most similar protein structure pairs have less than 12% pairwise sequence identity, and the average sequence similarity among all pairs of similar structures falls into 8% to 10% which defines a region known as the *midnight* zone (Rost, 1997). If pairwise sequence identities increase to 20-35% which was originally called *twilight* zone (Doolittle, 1986), sequence alignment methods often fail to align correctly protein pairs because pairwise sequence identity, the percentage of residues that are identical between two protein sequences, is not sufficient to define pairwise homology.

Later on, a length-dependent cut-off technique for significant sequence identity showed 95% of protein pairs have different structures in the twilight zone. On one hand, however, based on the length-dependent cut-off approach, over 90% of protein pairs that had been identified above the cut-off threshold were homologous, which roughly corresponds to 30% pairwise sequence similarity. On the other hand, less than 10% of protein pairs were homologous when the cut-off threshold corresponds to less than 25% pairwise sequence similarity (Rost, 1999). Therefore, the cut-off threshold corresponding to the pairwise sequence similarity of less than 25%, which is known as below the "twilight zone", is a restriction by which we can expect to identify protein pairs with the least structural similarities. The criteria of a protein dataset with pairwise sequence similarity below the twilight zone is very important in general for protein structure prediction methods, and specifically for PSS prediction. A learning PSS prediction model that incorporates a dataset with higher pairwise sequence similarity will be potentially overfitted such that it performs more effectively

on query sequences that are homologous with training sequences, and therefore, the prediction does not reflect throughly the method's performance. The effect of incorporating datasets with different levels of homology on the performance of a PSS prediction method is discussed in Chapter 5.

RS126 (Sander & Schneider, 1991) and CB513 (Cuff & Barton, 1999) are the two commonly used datasets whose pairwise sequence similarity is below the twilight zone (25%) and which contain a small and medium number of protein chains, 126 and 513 chains respectively. Due to the increase of protein sequences with known structures over time, more protein datasets have been compiled such as ASTRAL (Martin et al., 2006), EVA5 (Lin, Simossis, Taylor, & Heringa, 2005), PROTEUS-2D (Montgomerie et al., 2006), etc. These datasets may also contain sequences from the two standard datasets, and in other cases, protein datasets have been referenced by a set of chains' IDs. More centralized sources of protein databases with different levels of sequence similarity are protein culling servers such as PISCES (Wang & Dunbrack, 2003), and such datasets are updated frequently with new protein sequences, and for a performance comparison, it should be considered that two datasets with the same label may contain different numbers of protein chains. For an unbiased performance evaluation of two PSS prediction methods at least the same level of sequence similarity in the protein sequences used in learning and testing phases are necessary as was discussed earlier.

A common approach for the evaluation of PSS prediction is an n-fold cross-validation technique by which the average correlation coefficient values of the three classes of secondary structures are measured. In this study, we initially employ the same evaluation technique in the experiments. While these types of experiments give a measure of performance for individual approaches, we would like to know with confidence that one method performs better than another. Therefore, for a more

comprehensive performance comparison, we perform statistical tests and analyze the distributions of prediction scores. We suggest that future work in this area use statistical test to compare the performance of different systems.

1.6 Proposed Secondary Structure Prediction Model

In this study, we propose an *ab initio* PSS prediction model based on ML techniques and new amino acid encoding schemes in two stages: (1) a novel PSS transition site prediction model which incorporates the information derived from amino acid sequences by using a new technique based on a k-means clustering, Information theory and Ramachandran map (Ramachandran & Sasisekharan, 1968), and (2) a three-state PSS prediction using the statistical information derived from the PSS transition site prediction model in the first stage. The prediction model is based on a cascaded ML technique which is a common and effective architecture among state-of-the-art PSS prediction methods. An *ab initio* PSS prediction method incorporates the information derived from protein sequences and does not use directly the structural information of proteins.

In the first stage of the proposed PSS prediction model, in addition of PSS transition site prediction, the PSS transition site model provides the information that is incorporated in the second stage for secondary structure prediction. For a protein sequence, PSS transition sites are the locations on the sequence where one type of a secondary structure element is changed to another type of a secondary structure element. A detailed explanation of the proposed PSS prediction model is provided in Chapter 4.

The PSS prediction in two stages has several advantages: (1) the complexity of the PSS classifier is reduced by simplifying target features, (2) fewer dimensions in

input vectors of the secondary structure prediction model which reduce the curse of dimensionality, less trainable parameters and computation, (3) secondary structure transition sites which represent the overall topology of protein backbones at a secondary structure level are valuable information used in the PSS prediction model, and (4) the information of transition sites can be utilized in comparative modeling of protein structure determination by which a region on a protein sequence is assigned to a specific secondary structure, however, the extent of the secondary structure in both sides of the region cannot be determined.

Protein sequences are coded by various numbers and orders of twenty amino acids. This means that a learning model encounters an enormously large input space when the model incorporates protein sequences. Amino acid sequences and other sequence information have been used directly in PSS methods, however such approaches do not reduce the input space and ignore the intermediate structural information such as secondary structure transition sites.

The transition sites represent linearly structural information which can be identified from sequences and reduce the input space from 20^l to 2^l where l is the length of a sequence segment defined by a fixed-length sliding window. The proposed PSS prediction in two stages reduces the complexity of secondary structure prediction model due to less learning parameters and target features.

We evaluated the proposed PSS prediction model with the latest protein database, PISCES, from a protein culling server (Wang & Dunbrack, 2003). The dataset contains over nine thousand chains and approximately two million residues, and the sequence identity of every two sequences is less than 25%. The performance of the proposed two-stage PSS prediction method was compared to the state-of-the-art PSS prediction methods which commonly employ cascaded ANNs and SVMs. The experimental results and statistical analyses indicate a significant improvement in the

overall distribution of prediction accuracy with p-values less than 0.001.

In a standard approach, the performances of PSS methods are measured by the Q_3 and SOV scores, and Matthews correlation coefficient (MCC) is computed for the three secondary structures. Initially, in our experiments, we measured and compared the performances of the related methods by the common evaluation techniques.

However, we performed additionally a number of statistical tests and analyses for a more comprehensive comparison of the proposed PSS prediction model with the other methods in order to explore and analyze the distributions of the methods' prediction scores. It is recommended that the statistical approach to be adopted as a more comprehensive comparison technique for the performances of PSS prediction methods in other studies.

1.7 Thesis Organization

This thesis is arranged as follows. In Chapter 2, we review PSS prediction methods. Protein secondary structures and the importance of PSS prediction in protein tertiary structure determinations are illustrated. The common schemes of protein sequence representations are discussed in detail. The performances, limitations, strengths and variations of prominent PSS prediction methods are discussed.

In Chapter 3, several computational intelligence techniques, which are artificial neural networks, support vector machines, genetic programming, and clustering, are illustrated. These machine learning methods are employed in the proposed protein secondary structure prediction model, or they are used independently in the experiments for evaluations and comparisons. The concept, structure, learning algorithms and applications of the methods are explained in detail.

We review, additionally, protein structures and some related subjects such as the

different types, geometries and properties of secondary structures in a computational context, and the techniques that are used for secondary structure assignments. Amino acid encoding is an important step in ML methods for protein secondary structure prediction. We explain a number of amino acid encoding schemes. Lastly, two common evaluation techniques which are used in secondary structure prediction methods are discussed.

Our proposed two-stage protein secondary structure prediction model is illustrated in Chapter 4. The structure, components, data representations and prediction phases of the proposed model are described in depth and with exemplary figures and data.

In Chapter 5, we illustrate the various levels of experimental setups, and the experiments that are performed in this study. A number of common amino acid encoding schemes are introduced including the proposed amino acid encoding schemes. We explain our new approach which is based on substitution matrices for the evaluation of encoding schemes. The proposed Codon encoding scheme is evaluated further for PSS prediction by using different support vector machine architectures, and the effectiveness of the proposed encoding scheme is compared to those of commonly used amino acid encoding schemes.

The second group of experiments that are conducted in this study are related to the proposed GP model and a new technique to encode protein sequences by using clustering, Information theory and Ramachandran plot. The efficiency of clustering and performance of the GP prediction model in PSS prediction are compared to those of two commonly used machine learning methods which are artificial neural networks and support vector machines.

The third group of experiments are related to the proposed two-stage PSS prediction model. The final set of the experiments address the four aspects of the proposed two-stage PSS prediction model: (1) incorporating the information of PSS transition

sites, (2) the sizes of protein datasets, (3) the performance of PSS transition site model, (4) evaluation techniques and statistical analyses.

The proposed two-stage model is evaluated initially without incorporating PSS transition sites with two common datasets with small and medium sizes, and the conventional approach is used to generalized the results. In addition, we use the latest nonhomologous dataset with a very large size and performed a number of statistical tests for the performance evaluation.

In Chapter 6, the summary and contribution of this study are pointed out, and we outline the future work that can be conducted potentially to improve the performance and efficiency of the PSS prediction model developed in this thesis.

Chapter 2

Literature Review

2.1 Introduction

In this chapter, we mainly review protein secondary structure (PSS) prediction methods. Initially, protein secondary structures and the importance of PSS methods in protein 3D structure modeling are briefly explained. The commonly used techniques of protein sequence representations are discussed in detail.

The representation of protein sequences is the first step in all PSS prediction methods which has significant impact on overall prediction performances regardless of the methods' designs and learning parameters. In general, PSS prediction methods are grouped in four categories whose performances, limitations, and strengths are illustrated and compared in Section 2.2.2. In addition, PSS prediction methods with different architectures and those methods based on hybrid and consensus schemes are reviewed, and the performances and advantages of the methods are explained.

We discuss a number of common variations among PSS prediction methods and challenges in PSS prediction. Lastly, we point out a common difference among PSS

methods which has an impact on the evaluation and comparison of PSS prediction methods, and it is related to experimental setups and materials.

2.2 Secondary Structure Prediction Methods

Protein secondary structure (PSS) prediction is a fast and inexpensive way of decoding the tertiary structures of proteins which are essential to understand the biological functions and mechanism of proteins. PSS prediction is an alternative approach to defining guiding points for modeling proteins' 3D conformations primarily from their amino acid sequences.

A promising PSS prediction method can be utilized to reduce the complexity of tertiary structure prediction methods, and help interpreting empirical data obtained by experimental methods for protein structure determination when accurate or complete structural information is not available. Protein secondary structures are the spatial patterns observed in protein 3D structures with regards to the hydrogen bonds among adjacent or distant residues, and these patterns are grouped mainly in three types which are helices, strands, and coils. Hydrogen bonds are formed between amide hydrogens and carbonyl oxygens as shown in Figure 2.1. A detailed review of protein secondary structures is provided in Chapter 4.

2.2.1 Early PSS Prediction Methods

The early PSS prediction methods were based on single-residue statistics that aimed to find the correlation of certain amino acids with an α-helix, and the concept was improved over time to explore the preference of all amino acids that have significant roles in an α-helix or β-strand formation (Szent-Gyorgyi, Cohen et al., 1957; Prothero,

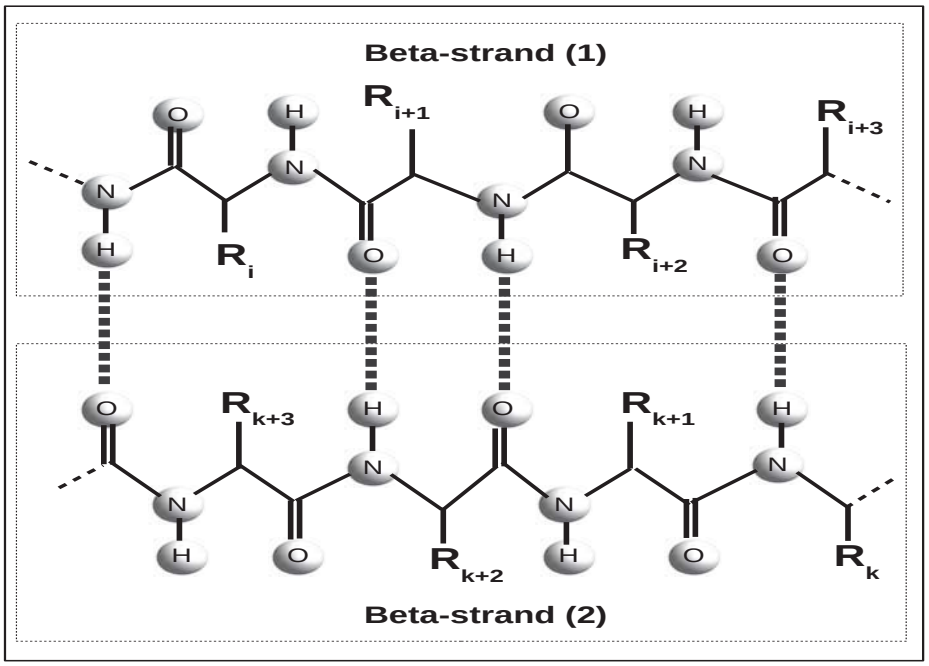

Figure 2.1: An schematic overview of a β-sheet formed by two beta-strands which are connected by the hydrogen bonds of distant residues denoted to R_i and R_k $(i, k = 1, \cdots, n)$.

1966; Periti, Quagliarotti, & Liquori, 1967; Finkelstein & Ptitsyn, 1971; Chou & Fasman, 1974b; Lim, 1974b; Chou, Fasman et al., 1978; Garnier et al., 1978).

For instance, in the Chou & Fasman method, the propensity values of residues and the empirical rules of helix formations were computed and formulated respectively from limited protein sequences with known structures. Then, regions with higher possibilities of helical formations were predicted as follows (Chou & Fasman, 1974b):

1. Propensity values of helix and strand, p_α and p_β respectively, are assigned to all residues in a sequence. The propensity values are computed based on the

2. If the p_α values of four out of six consecutive residues are higher than 1.0 then the residues are grouped and temporarily marked as a helix forming region.

3. Each detected helical region is extended from both directions until the average p_α values of four consecutive residues is less than 1.0.

4. For each identified region that has more than five residues if the average p_α is greater than the average p_β within the region, then it is labeled as a helix.

A drawback for the aforementioned class of PSS methods was that the correlation information was extracted from a limited number of proteins with known structures. Therefore, the reported prediction accuracy at the time did not reflect a decisive evaluation of the methods' performances. A decade later when more protein structures became available, it was discovered that the stated three-state accuracy (Q_3) had been overestimated, and the accuracy scores reached at most 56% (Kabsch et al., 1983).

An improvement to PSS prediction emerged when the correlations of amino acid preferences for particular secondary structures were extracted from the segments of protein sequences. An amino acid segment is defined by a number of consecutive

residues which is usually from 11 to 21 residues. Then, the likelihoods of the middle residues in amino acid segments are evaluated for participating in a particular secondary structure (Ptitsyn & Finkelstein, 1983; Gibrat et al., 1987; Schneider, 1989; Garnier et al., 1996). The three-state prediction scores of the methods based on the information of amino acid segments were improved, in average slightly over 60%, compared to those of the methods based on the preference of a single residue for particular secondary structures.

There are two common drawbacks for the methods based on single or multiple residue information: (1) the overall Q_3 scores are unacceptably low, specifically low prediction accuracy for β-strands which range from 28% to 48%, slightly better than random secondary structure assignments, and (2) the short lengths of predicted strands and helices (Rost & Sander, 2000). These problems originate from the facts that distant residue interactions play important roles in secondary structure formations in which these long-range contacts are not captured sufficiently even with the information of neighbouring residues in amino acid segments.

A major improvements in PSS prediction accuracy emerged when protein evolutionary information was incorporated (Levin, Pascarella, Argos, & Garnier, 1993; Solovyev & Salamov, 1994; Rost et al., 1993). It has been discovered that multi sequence alignments of homologous proteins contain the information of long-range residue contacts.

The underlying fact for the improvement is that if two protein structures are similar, the mutation of one or two residues at two distant locations occurs with residues that preserve the physicochemical properties of the original residues, and consequently preserving the protein structure (Lichtarge, Bourne, Cohen et al., 1996). This phenomenon is enforced by evolutionary pressure, otherwise mutations would cause structural changes that are destructive.

As a result, searching for evolutionary relations reveals the residues that are functionally important with regards to the relation between the function and structure of a protein. Protein evolutionary information is more decisive by two observations: (1) all pairs of homologous protein sequences have similar structures if they have at least 35% identical pairwise residues over one hundred aligned residues (Sander & Schneider, 1991; Rost, 1999), and (2) a majority of protein pairs with similar structures have less than 15% pairwise sequence similarity (Rost, 1997). Therefore, the level of pairwise sequence similarity which is considered in a PSS prediction method is decisive to the performance evaluation and generalization of the learning method.

2.2.2 Recent PSS Prediction Methods

In a general scheme, protein secondary structure methods fall into four categories:

1. Stereo- and physico-chemical properties of amino acids
2. Statistics and Information theory
3. Sequence homology
4. Machine Learning (ML)

In protein secondary structure methods based on the stereo- and physico-chemical properties of amino acids, residues in protein sequences are represented by their physiochemical properties and empirical rules that govern the formation of α-helix and β-strand are defined based on the various grouping and orders of amino acid properties such as mass, hydrophobicity, hydrophilicity, charge, etc (Lim, 1974a).

In statistical approaches based on Information theory and Bayesian statistics the correlation between the amino acid sequence and secondary structures of a protein

are formulated. In other words, these methods compute the preference of each amino acid for a given secondary structure element (Garnier et al., 1978; Gibrat et al., 1987; Garnier et al., 1996; Kloczkowski et al., 2002).

The third group of protein secondary structure methods assign the secondary structures of a sequence based on sequence homology. By these methods, a protein database is searched for detecting the homology between the fragments of a target sequence and the sequences of a protein database. These methods are based on the hypothesis that short segments of protein sequences with distant homology can have possibly similar secondary structures (Nishikawa & Ooi, 1986; Levin et al., 1986). All regions (short segments) of protein sequences are searched to identify a set of amino acid segments with the highest similarity match to a target segment by using a substitution matrix.

In a protein database, all sequences are scanned by a fixed-length sliding window which moves one residue at a time along sequences. The segments that have a total sequence similarity greater than a predefined threshold are noted in a matrix $n \times m$ where n is the length of a target sequence and m is the number of secondary structure states. The elements of the matrix at each column are decreased or increased by a certain percentage based on the periodicity of each secondary structure to prevent possibly an under or over prediction outcome due to the unbalanced numbers of observed secondary structures. Under or over prediction is referred to a prediction result which is often biased toward a specific secondary structure due to uneven distribution of the three types of secondary structures. Finally, in the matrix a column with a maximum value defines the secondary structure of a residue in each row.

The last category of protein secondary structure methods are based on ML techniques which employ artificial neural networks (ANNs) (Qian & Sejnowski, 1988;

Holley & Karplus, 1989; Rost & Sander, 1994a; Kneller, Cohen, & Langridge, 1990; Zhang, Mesirov, Waltz, & Cohen, 1992; Jones, 1999a; Chen & Chaudhari, 2007), support vector machines (SVMs) (Hua & Sun, 2001; Nguyen, Rajapakse et al., 2003; Guo, Chen, Sun, & Lin, 2004), and hidden markov models (HMMs) (Asai et al., 1993; Karplus, Sjölander, Barrett, Cline, Haussler, Hughey, Holm, Sander et al., 1997; Lin et al., 2005).

2.2.3 PSS prediction based on Statistics and Information theory

The early developments of PSS prediction methods were based on the statistical analysis of protein databases and Information theory (Shannon, Weaver, Blahut, & Hajek, 1949). By using Information theory, one can compute the information contained in a set of symbols which have different probabilities *a priori*. The information I which an event y carries on the occurrence of an event x, or a measure of statistical constraint between two events x and y, is formalized as the following (Robson, 1974):

$$I(x;y) = \log \frac{p(x|y)}{p(x)} \tag{2.1}$$

where $p(x)$ is the probability of event x and $p(x|y)$ is the conditional probability of x, knowing event y has happened. If events x, y are independent, $I(x;y)$ is equal to 0 according to Equation (2.1). The value of $I(x;y)$ will be greater or less than 0 if the occurrence of event y supports or does not support the occurrence of x respectively.

In a protein sequence with n residues, event $x{=}S_i$ is defined as one of the common secondary structures that a residue at location i adopts, and the formation of a secondary structure depends on event $y{=}R_j$ which is defined as the type of the residue at location j. Ideally, S_i is dependent on all residues in a protein sequence. The information of event S_i associated to a residue at the location i is formalized as

follows:

$$\begin{aligned} I(S_i; R_1, R_2, \ldots, R_n) &= \log \frac{p(S_i|R_1, R_2, \ldots, R_n)}{p(S_i)} \\ &= I(S_i; R_i) + I(S_i; R_{i-1}|R_i) + I(S_i; R_{i+1}|R_i) \\ &\quad + I(S_i; R_{i-2}|R_{i-1}, R_i) + I(S_i; R_{i+2}|R_{i+1}, R_i) \\ &\quad + I(S_i; R_{i-3}|R_{i-2}, R_{i-1}, R_i) + \ldots \end{aligned} \tag{2.2}$$

where x_i is referred to the three secondary structures (helix (H), strand (E) and coil (C)). Thus, the ith residue is assigned to a secondary structure that has the highest information value. The information term formulated in Equation (2.2) is computed by expanding it to the sum of n+1 simpler pieces of information. Additionally, in Equation (2.2) the terms that are dependent on the combinations of more than three residues are only limited to three residues since the values of the statistical information of the terms with the combination of more than three residues will be equal or close to zero.

Moreover, it was observed that the type of a residue, e.g. at the ith location, has a significant role on the secondary structures of up to eight consecutive residues in both directions (N and C-termini) of the residue. Therefore, the left-hand information term of Equation (2.2) can be simplified as follows:

$$I(S_i; R_{i-8}, R_{i-7}, \ldots, R_{i+7}, R_{i+8}) \tag{2.3}$$

where S_i is the secondary structure of the ith residue located in the middle of an amino acid segment. If the information term I is computed from a sequence segment with less than sixteen residues, it results in neglecting some significant information related to the potential effects of neighbouring residues. Also, the interactions of two residues that are located farther than eight residues from each other are considered insignificant (Robson & Suzuki, 1976).

Moreover, if the information term I is expanded to simple information terms which are dependent only on the combination of a single residue, the number of parameters that are required to compute the simplified information term I is equal to $20\times17\times3$ where 20 is the number of amino acids, 17 is the number of residues in a selected sequence segment and 3 is the number of secondary structure states.

2.2.4 PSS prediction based on Machine Learning

Machine Learning (ML) techniques have been proved to be capable of solving many problems in the fields of engineering and science. Hence, a significant achievement in PSS prediction has occurred by employing ML methods. In this section, we outlined a number of important methods using various ML schemes.

Artificial Neural Networks (ANNs)

In the ML category, early study of PSS prediction was based on a fully connected Multilayer Perceptrons using the Backpropagation (BP) learning algorithm (Rumelhart, Hinton, & Williams, 1986). Artificial Neural Networks (ANNs) have been applied for learning of "text to speech" where strings of words and phenoms are the input and output of the neural networks respectively. Likewise, in the context of protein secondary structure, amino acid sequences (strings) and their corresponding secondary structures are the input and output of the neural networks. In this approach, protein sequences are split into a set of amino acid segments by moving a fixed-length sliding window, one amino acid at each time, along a sequence. The amino acids in a segment are the inputs of the neural network (Qian & Sejnowski, 1988).

Moreover, the inputs are encoded from symbolic representations to numeric values in order to be processed by a neural network. The output of the neural network is

one of the three classes of protein secondary structure corresponding to the central residue of an input segment. A common encoding scheme encodes each amino acid or a spacer symbol by a 21-dimensional orthogonal vector (Swanson, 1984). The start and end of a protein sequence are marked by a spacer symbol. Each dimension (unit) corresponds to a spacer or one of the twenty amino acids if the unit is set to 1 and the rest of the units are set to 0. The overall prediction accuracy of the method based on the three-state score, Q_3, was approximately 64%.

Although, the proposed technique improved the prediction accuracy, it could not detect effectively the periodicity of the α-helices and β-strands. A number of modifications to the inputs of the neural networks were proposed that improved slightly the prediction of regular secondary structures such as adding one real value to represent the hydrophobicity of the central amino acid (Kneller et al., 1990), and four extra units to the orthogonal binary vector (Sasagawa & Tajima, 1993) for the symbols *B, X, Z*, and chain breaks which are assigned to certain residues according to the dictionary of protein secondary structures (Kabsch & Sander, 1983).

In addition, a parallel hybrid model using a Genetic Algorithm (GA) (Holland, 1992) and an ANN was proposed which had the advantages of a robust search for exploring the parameter space of the neural network and selecting optimal neural network topologies (Vivarelli, Giusti, Villani, Campanini, Fariselli, Compiani, & Casadio, 1995). By the hybrid approach it has been shown that the performances of neural networks with various architectures and different values of adjustable parameters are only slightly different. In other words, the experimental results indicate there is a limitation to improving the prediction accuracy of the neural networks with different topologies and learning algorithms (Dongardive & Abraham, 2015) if the inputs of the neural networks are only limited to amino acid sequences as described.

Meanwhile, a more complex hybrid PSS model based on three predicting compo-

nents was proposed as follows (Zhang et al., 1992): (1) a memory-based reasoning component which is based on the concept of a homology-based method. Using a distance function, it identifies the nearest-neighbour segments which are homologous to the query segment from all available training sequences, (2) a statistical component that computes the probability of a secondary structure for a central residue in a target amino acid segment, and (3) a feed-forward neural network.

In the method, for the central residue of a query segment each predicting component generates three scores related to the three secondary structures. A *combiner* module which is also a neural network takes $n \times 9$ scores as inputs to perform the final prediction where n is the number of residues (sliding window's length) in a query segment. The hybrid model has improved the overall Q_3 score up to 66%. However, a common drawback of the earlier protein secondary structure (PSS) methods was the sole use of local information which is referred to the order and type of a predefined number of adjacent residues in a protein sequence that affect the formation of secondary structures, specifically helices, as a whole.

The performance evaluation of PSS prediction methods based on various statistical and neural network approaches indicates that only using the local information of amino acid sequences is not enough to improve the classification accuracy even with different neural network topologies, learning parameters and various inputs. In fact, a prediction method that uses solely the local information of amino acid sequences cannot sufficiently capture the correlation information among residues and consequently hinders further improvements to classification performance.

A major breakthrough in PSS prediction was achieved when protein evolutionary information was incorporated with cascaded neural networks such PHD (Rost et al., 1993; Rost, 1996; Przybylski & Rost, 2002) and JNET (Cuff et al., 2000). In the study, two tiers of neural networks were employed for processing inputs. In the first

tier, the sequence-structure network whose inputs are the occurrence frequencies of amino acids are computed for each sequence through a fast homology search such as BLAST (Altschul, Gish, Miller, Myers, Lipman et al., 1990; Altschul, Madden, Schäffer, Zhang, Zhang, Miller, & Lipman, 1997) in a protein database. In the second tier, the inputs of the structure-structure network are the outputs of the first neural network.

The outputs of the first neural network are three real values representing the tendency of each central residue to be in the secondary structures: strand, helix and coil. A central residue is located in the middle of an amino acid segment which is defined by a sliding window that moves by one residue at each time along a protein sequence. The first network learns sequence to structure information which is the correlation between the secondary structure of the central residue and the types of neighbouring residues within a sliding window while the second network learns structure to structure information which is the correlation between the secondary structure of a central residue and the secondary structures of adjacent residues within the sliding window. Unlike local information, correlation information is referred to the location and type of residues that are not necessarily adjacent and have decisive roles in the formation of secondary structures, specifically for β-sheets that are mostly dependent on long-range residue interactions in a protein sequence.

The frequencies of residues from a multi-sequence alignment are computed as follows. For example, let's assume a multi-sequence alignment is generated with nine sequences homologous to a target sequence and the ith residue in the target sequence is the amino acid "D". If the ith amino acid of the target sequence is aligned with the amino acids "KEDDEDDKE" at the ith column of the multi-sequence alignment as shown in Figure 2.2, the computed frequencies of the amino acids "D", "E", "K" are equal to 0.5, 0.3 and 0.2 respectively in the example. The cascaded

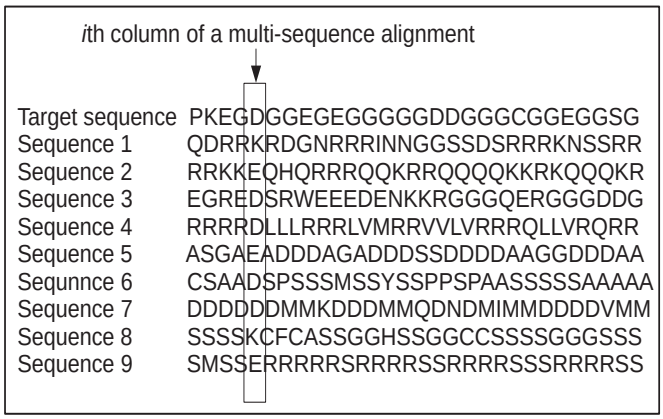

Figure 2.2: An example of a multi-sequence alignment to compute the frequencies of amino acids at location i on a target sequence by using template sequences.

neural network approach was tested on the dataset RS126 (Sander & Schneider, 1991), and an overall Q_3 score of above 70% was achieved using a 7-fold cross-validation technique. The dataset is one of the standard datasets for the evaluation of the PSS prediction method in which every two sequences with more than 80 residues have a pairwise similarity of less than 25%. The length-dependent cutoff defined by homology-derived secondary structure of protein (HSSP) (Sander & Schneider, 1991) filters out the sequences with a significant pairwise sequence similarity which are homologous proteins and prospectively with structural similarities.

The cut-off rule for limiting sequence similarity is important since profile information derived by alignment procedures for a training sequence can be also obtained for a test sequence if the two sequence have a significant homology. Also, to improve prediction accuracy, the sequence homology of sequences in an alignment with a target sequence should cover a wide range from 30% to 90%. Alternatively, global information can be incorporated with the neural networks such as, amino acid composition, lengths of sequences and the distance of a central residue to C and N-termini.

PSS prediction methods that use evolutionary information from multi-sequence alignments are faced with two challenges: (1) the compilation of alignments is computationally expensive and it requires powerful computing systems, and (2) the prediction accuracy relies on the divergence of sequences in an alignment due to the evidence that suggests including sequences with low sequence similarity increase prediction accuracy in comparison to choosing only sequences closely related to a query sequence. In other words, if protein profiles are generated from multi-sequence alignments that include solely protein sequences with high sequence similarity, a PSS method that incorporates the profiles does perform well only on query sequences that have high sequence similarity with protein sequences from which the profiles are derived.

PSIPRED (Jones, 1999a) is an alternative method that employs two cascaded neural networks and it does not require multi-sequence alignment procedures which overcomes the challenges related to sequence alignments. The method searches protein databases for sequence homology by PSI-BLAST (Altschul et al., 1997) which is more efficient to detect distant homology compared to the standard Smith-Waterman method (Smith & Waterman, 1981).

The advantage of the PSI-BLAST search is that new profiles of sequences are simultaneously generated from multi-sequence alignments (MSAs) while searching a protein database. Therefore, in each iteration, an MSA is formed by using the newly generated profile. In PSIPRED, for a target sequence a position-specific scoring matrix (PSSM) is generated after three PSI-BLAST iterations. An element a_{ij} of the PSSM is the log-likelihood that the amino acid i is at position j of the MSA. Thus, the PSSM contains $20 \times n$ entries where 20 represents the twenty amino acids and n is the length of the target sequence.

The PSIPRED method was evaluated on 187 protein sequences with 3-fold cross-validation technique and the 21 target sequences of the third critical assessment of

methods of protein structure prediction (CASP) (Moult, Hubbard, Bryant, Fidelis, & Pedersen, 1998), and an average Q_3 score of 76% was reported.

Although profiles generated by PSI-BLAST search increase the performance of a prediction method, a redundant database can negatively change prediction outcomes. In such a case, if repetitive sequences are incorporated in an intermediate profile, random sequences are matched and consequently the search method generates incorrect results.

A drawback of prediction methods using short segments of adjacent amino acids is that the methods are not able to intertwine the information of local amino acids from short segments with that of distant amino acids from the entire sequence. In order to incorporate, in certain degree, the correlations among local and nonlocal amino acids, bidirectional recurrent neural networks (BRNNs) were interconnected in a special neural network topology by SSpro (Baldi, Brunak, Frasconi, Soda, & Pollastri, 1999; Pollastri, Przybylski, Rost, & Baldi, 2002; Pollastri & Mclysaght, 2005; Mirabello & Pollastri, 2013).

The model consists of a standard feed-forward neural network and two similar recurrent networks at the left and right of the standard neural network. A segment of amino acids for whose central residue the prediction is performed is fed to the middle neural network, and at a time t, the inputs of the two recurrent networks are their outputs at time t-1. In this way, the recurrent neural networks at left and right sides interpreted, as *wheels*, make the correlations among amino segments of an entire sequence from *c*-terminus to *n*-terminus, and in other words, incorporate long range information.

The entire model is trained by the Backpropagation algorithm through time. SSPro method performance was tested using BLAST and PSI-BLAST profiles and the latter showed a better performance. In the method, all protein sequences were

extracted from the protein data bank (PDB) (Sussman et al., 1998) and after applying redundancy and homology checks, 1180 sequences were selected for training. The train models were tested on RS126 (Rost et al., 1993), EVA (Cuff & Barton, 1999) and the CASP dataset which consist of 126, 223 and 40 sequences respectively. SSPro showed a relative performance improvement with an average Q_3 score of 77%.

PORTER is a computer program based on the SSpro method which has been developed as an automated server with more precise coding of input profiles from multi-sequence alignments and updated when new protein sequences with known structures become available (Pollastri & Mclysaght, 2005). In addition to protein evolutionary information, the relative solvent accessibility (RSA) of amino acids has been also observed to be effective when incorporated in a PSS prediction method, and informative about protein structures. RSA is the degree to which a residue in a protein is accessible to a solvent molecule and indicates a fingerprint of the overall topology of a protein structure (Rost & Sander, 1994b).

In the SABLE method (Adamczak, Porollo, & Meller, 2005), the RSA information in the form of real values or two classes (exposed if RSA>25% and buried if RSA$\leq$ 25%) was obtained in two ways to examine its effect on a PSS prediction method : (1) the RSAs computed from experimental data, and (2) the RSAs acquired by an RSA prediction method. In SABLE, cascaded neural networks with two hidden layers are used in two stages, similar to the neural network architecture in the PHD methods. However, the outputs of two hidden layers are returned to themselves through designated hidden layers called *context units*.

The context units act as a temporal memory for prior states of internal units, and the connection weights of context units are not adjusted during training since the outputs of context units were designated to add various degrees of excitations to hidden layers based on training sequences that were presented at earlier times.

The experimental results indicated the prediction accuracy of RSAs increased the previously obtained Q_3 and segment overlap scores by 3% to 5%.

Support Vector Machines (SVMs)

In the next group of PSS prediction methods based on ML approaches, support vector machines (SVMs) were employed with various designs. For instance, a PSS prediction method based on binary SVMs was examined to improve the prediction performance (Hua & Sun, 2001). In the study, secondary structure prediction was performed in two phases. Initially, six binary SVM classifiers were trained based on the two consensus schemes:

1. One-against-all (OAA).
2. One-against-one (OAO).

By the first scheme, three binary classifiers were constructed denoted as (H/∼H), (E/∼E) and (C/∼C). Each binary classifier was trained to classify one secondary structure versus the two other secondary structures. For example, the classifier (H/∼H) discriminates helices versus strands and coils. By the second scheme, three binary classifiers are also constructed denoted as (H/E), (H/C) and (E/C). In the scheme, each binary classifier distinguishes one type of a secondary structure from another type of a secondary structure. For example, the classifier (H/E) classifies a helix against a strand.

As described earlier, training data was prepared similarly to the methods based on neural networks. Initially, amino acid sequences were segmented by a fixed-length sliding window and the amino acids defined by the window at each time were encoded as inputs, and the prediction was performed for the central residue of an amino acid segment.

In order to conduct a three-state classification, in the second phase tertiary classifiers were constructed by various combinations of aforementioned binary classifiers. For example, using OAA or OAO scheme, the prediction was performed by computing the outputs of all three binary classifiers for a target residue. The output of a binary classifier with the largest positive distance to optimal separating hyperplane (OSH) determined the predicted secondary structure of the residue.

More complex tertiary classifiers were constructed by cascading binary classifiers from the two binary classification schemes based on a directed acyclic graph (DAG). For example, if a cascaded tertiary classifier employs two binary classifiers (H/∼H), (E/C) in the first and second tiers respectively, when the OSH value of classifier (H/∼H) is negative, the final prediction is determined by the OSH value of classifier (E/C), and if the OSH value of classifier (H/∼H) is positive, the central residue is assigned to a helix.

In protein secondary structure prediction, the length and configuration space of protein sequences are very large, and it has been shown that choosing the radial basis function (RBF) as a kernel function results in a better performance than that of the other kernel functions in high dimensional spaces. The SVM-based PSS prediction method was evaluated by protein profiles derived from multi-sequence alignments, and average Q_3 of 71%, 73% and segment overlap (SOV) of 74%, 76% were achieved on the datasets RS126 and CB513 (Cuff & Barton, 1999) respectively a using seven-fold cross-validation technique.

The experimental results indicate a performance improvement compared to those of the PSS prediction methods employing cascaded neural networks. The overall prediction accuracy was increased due to the aggregation of all prediction from the tertiary classifiers using a consensus technique.

Another variation of SVM-based PSS prediction techniques is SVMpsi (Kim &

Park, 2003). The prediction method employed the combinations of tertiary SVM classifiers based on DAG schemes and incorporated PSI-BLAST profiles. The method was evaluated on RS126 and CB513 sequences using a seven-fold cross-validation technique, and an average Q_3 of 76% and segment overlap (SOV) of 73% were achieved respectively.

Moreover, the method resulted in an average Q_3 of 73% and segment overlap (SOV) of 72% when the multi-sequence alignment profiles of the datasets were used in the experiments. The experimental results indicate a higher Q_3 score, approximately 3%, compared to those of SVM-based prediction methods incorporating multi-sequence alignment profiles.

Alternatively, multi-class SVMs and cascaded multi-class SVMs were studied additionally for PSS prediction (Nguyen et al., 2003). In a multi-class SVM method, three discriminant functions are constructed to perform a three-state classification which is contrary to the classification that is performed by binary SVMs (Weston & Watkins, 1999; Crammer & Singer, 2002). In (Nguyen et al., 2003), cascaded SVMs were employed separately for both binary and multi-class SVMs in two stages. In the first stage, the inputs of SVMs were the information of amino acid segments defined by a fixed-length sliding window, and the outputs of the SVMS were combined and represented to the SVMs in the second stage.

The cascaded SVM models were more effective at capturing the contextual variations among the residues and their long-range interactions. The experimental outcomes showed using a multi-class SVM with multi-sequence alignment profiles on RS126 sequences increased approximately 0.5% the average Q_3 compared to those of binary classifiers combined with a DAG model. Also, the average Q_3 increased by 2% if the architecture of the prediction model is enhanced from a multi-class SVM to cascaded multi-class SVMs with PSI-BLAST profiles.

Hidden Markov Models (HMMs)

Protein secondary structure prediction has been also studied using Hidden Markov Models (HMMs) (Asai et al., 1993; Karplus et al., 1997). In the studies, the eight secondary structures defined by DSSP are grouped in helix, turn, strand and coils, and four HMMs are defined with a number of hidden states and their state transition connectivities. The topology of each HMM model is based on the empirical rules governing on the four classes of the defined secondary structures. For example, a helical scondary structure is formed by at least three consecutive resides, and therefore, the HMM structure related to a helix contains three hidden states with state transition loops on them. Each HMM corresponds to one of the secondary structures and is separately trained by the Baum-Welch algorithm and then the four HMMs are combined (or connected to an initial state) for an overall prediction.

In the model, observations are amino acid sequences and the hidden states that lead to maximum probability for observed sequences determine the predicted secondary structures. To prepare training data for the four HMMs, training sequences are segmented according to the start and end of their labeled secondary structures. The amino acid segments with similar secondary structures are grouped in four sets and used separately to train the HMMs. In the method, an observed symbol is one of the twenty amino acids.

In order to capture more effectively the relations among neighbouring amino acids, an output (possible observation) symbol can be represented by two amino acids. Therefore, the number of output symbols increases to 400. If the number of output symbols is increased, it adds more complexity due to the increase of training parameters related to emission probabilities. Nonetheless, based on the experimental results using a two- or three-letter scheme for output symbols improves prediction accuracy if available sequences are enough to evaluate two- or three-letter correlations. The

HMM model was evaluated by a protein dataset that consists of 120 sequences without incorporating alignment profiles. The jackknife estimation technique was used for generalization and it achieved an average Q_3 of 66%. The jackknife or "leave one out" procedure is a cross-validation technique which is usually used if a dataset does not contain a sufficient number of samples.

Constructing a PSS prediction technique based on the HMM method highly depends on a defined HMM topology, the number of states and their connectivities, and it has a great impact on the performance of the HMM model. Using an evolutionary computation such as Genetic Algorithm (GA) helps exploring different HMM topologies and improving structure prediction (Won, Prügel-Bennett, & Krogh, 2004; Won, Hamelryck, Prügel-Bennett, & Krogh, 2007). In PSHMM (Won et al., 2007), initially four block-HMMs, linear, self-looped, forward-jump and zero, and the genetic operators for the modifications of the block-HMMs are defined. Then, a GA is applied on a population of HMMs which are composed of randomly generated and connected block-HMMs.

In PSHMM, the average Q_3 scores of 68.3% and 75.1% were reported by incorporating single and multiple sequence information on the 2230 sequences of the SABMark dataset. In an alternative approach, the OSSHMM method constructs various HMM topologies and explores an optimal HMM model based on an automated selection of HMM topologies (Martin, Gibrat, & Rodolphe, 2005a; Martin et al., 2006). In the method, without any prior knowledge a simple HMM with three components, each including only one state, related to three secondary structures is constructed. The initial HMM size is increased by adding progressively states and parameters, and a statistical information criterion and a symmetric distance measure are used to compare different HMMs and select an HMM which performs well on a dataset.

The OSSHMM method achieved average Q_3 scores of 67.9% and 75.5% with single

sequences and multi-sequence alignments profiles respectively on 2524 sequences of the ASTRAL dataset using a 4-fold cross-validation technique. Moreover, a hybrid methods that consists of a neural network and an HMM has been studied to be effective for PSS prediction (Lin et al., 2005).

In YASPIN, similar to PSS prediction methods that employ cascaded-neural networks, a prediction is performed in two steps by a single-layer feed-forward neural network at the first stage and an HMM at the second stage. The neural network has seven outputs corresponding to seven types of secondary structures predicted for the central residue of a segment window with 15 residues. In the second stage, the neural network outputs are used in a single HMM to perform a final three-state structure prediction.

A new aspect in the method is that a helix structure is defined by three classes (labels) corresponding to the *start*, *middle* and *end* positions in the helix and similarly for a strand structure. In this way, with the assumption of the correct predictions of *start* and *end* positions, the fragmented positions in the middle can be corrected. The experimental results on the 4256 sequences of PDB25 dataset indicate a comparable performance compared to well examined predicting methods such as PHD, SSPro and PSIPRED methods.

Consensus and hybrid schemes

A number of PSS prediction models employ multiple techniques to improve the prediction performances. The k-nearest neighbour techniques with neural networks were shown to be effective for PSS prediction (Yi & Lander, 1993; Salamov, Solovyev et al., 1995; Salamov & Solovyev, 1997). As discussed previously, in the k-nearest neighbour technique the secondary structure of a central residue in a target segment is determined by searching all segments of training sequences which are homologous to the

target segment.

In this way, k training segments that have the highest sequence similarity with a target segment are selected. A scoring system based on the local environment scoring method is used to measure sequence similarity (Bowie, Luthy, & Eisenberg, 1991). By the local environment scoring method, initially every residue in the training sequences is assigned to an environment class which is determined by three features, the solvent accessibility, polarity and secondary structure.

In the method, three types of secondary structures and six types of polarity/solvent accessibility are considered, and therefore, a residue in every position is assigned to one of the eighteen environmental classes. Using Bayesian statistics and environmental sequences, represented by environmental classes, the score of a residue R_i to be observed in a structural environment E_j is calculated as follows:

$$score(E_j, R_i) = \log_{10}\left(\frac{p(R_i|E_j)}{p(R_i)}\right) \tag{2.4}$$

where $p(R_i)$ is the probability of residue R_i observed in any structural environment. In this way, a 18×20 environment scoring matrix is constructed and each element of the matrix is computed by Equation (2.4). Next, the 3D structure profile of a target sequence is constructed based on the scoring matrix. The 3D structure profile is a table with n rows and 18 columns where n is the length of the target sequence and each column corresponds to one of the 18 environment classes. A training sequence is scored by using the 3D structure profile of a target sequence, and the score measures the similarity of the two sequences.

Similarly, k segments with highest similarity to a target segment are determined and the most observed secondary structure in k neighbouring segments is assigned to the central residue of the target segment. Moreover, instead of choosing the maximum observed secondary structure as a prediction, alternatively the total numbers

of the three secondary structures in k neighbouring segments are computed, and the prediction is represented as a triplet.

In (Yi & Lander, 1993), the prediction performance was evaluated by 110 protein sequences with a 3-fold cross-validation technique. In this way, the output of the k-nearest neighbour method, which is a triplet corresponding to the central residue in a segment, was incorporated to a neural network for training. Therefore, the input units of the neural network are equal to $3 \times l$ where l is the length of the sliding window.

In addition, the k-nearest neighbour method was also investigated by incorporating evolutionary information from multi-sequence alignments (Salamov et al., 1995). In the study, instead of computing only the single homology score of a segment q in the training database and a target segment p, the homology scores of the segment q with all homologous segments of the segment p, derived from the multi-sequence alignment of the target sequence, are computed and the average of the homology scores determines the final homology score of the two segments p and q. The proposed nearest-neighbour technique achieved an average Q_3 of 72% using protein evolutionary information from multi-sequence alignments and a jackknife estimation technique on the RS126 dataset. The prediction accuracy is 4% higher than that of the similar method using single sequences.

Moreover, PSS prediction has been improved by incorporating directly the information of protein structures. As a consensus approach, the PROTEUS technique (Montgomerie et al., 2006) employed three prediction methods based on neural networks and structural alignments of homologous sequences with known structures to predict the secondary structure of a target protein as follows: (1) a BLAST search is performed for the target sequence to find homologous sequences, (2) three predictions are performed separately by the three PSS programs, JNET (Cuff et al., 2000), PSIPRED (Jones, 1999a) and TRANSSEC (Montgomerie et al., 2006), and

the three predictions are given to a jury system, which is also a neural network, to determine the final prediction of the three programs, (3) a multi-structure alignment is performed with the homologous sequences, and based on the alignment a 3D to 2D mapping is performed by a homology modeling to assign secondary structures as accurate as possible to the target sequence, and (4) the secondary structures obtained from the jury system and 3D to 2D mappings are aggregated for final prediction.

By the PROTEUS method, a Q_3 score of 79.1% was reported by using 100 sequences chosen randomly from the PROTEUS-2D dataset. In PROTEUS, the accuracy of 3D to 2D mappings relies on homologous sequences detected by a BLAST search. A mapping accuracy between 75% to 80% can be obtained if sequence similarities are between 25% to 40%.

SPINE (Dor & Zhou, 2007) is another PSS prediction technique based on a consensus technique. The SPINE method employs two separate sets of cascaded neural networks which have been used for three-state residue solvent accessibility (RSA) and three-state secondary structure predictions. The outputs of the two separate cascaded neural networks are integrated based on a voting scheme. The architecture of SPINE is the extended model of the commonly used two-stage neural networks proposed in the PHD method. The SPINE method was evaluated by incorporating the PSI-BLAST profiles of the CB513 dataset, in addition to 7 amino acids' properties, and an average Q_3 score of 76.7% was reported using a 10-fold cross validation technique.

A protein backbone can be represented by a set of dihedral angles which are computed by using the coordinates of C_α, C and N atoms as explained in Chapter 4. A dihedral angle is formed with two intersecting planes. For instance, by four consecutive atoms A, B, C and D in a 3D space, atoms A, B and C, and atoms B, C and D form two intersecting planes whose dihedral angle θ can have a value within

$[-180, +180]$. It has been previously studied that dihedral angles of a protein backbone are in certain ranges with regards to the types of local structures, and therefore, the values of a dihedral angle pair (ϕ,ψ) are not chosen randomly. The dihedral angle property has been used to predict discretized dihedral angles of protein structures from protein sequences (Zimmermann & Hansmann, 2008; Kountouris & Hirst, 2009).

The SPINEX method (Faraggi et al., 2012) is based on the fact that the prediction of dihedral angles of a protein backbone can improve PSS prediction. In contrary to previous studies based on a multi-state approach that predicted discretized dihedral angles (Zimmermann & Hansmann, 2008), the SPINEX method employs 10 cascaded neural networks whose topologies are similar to SPINE's in two levels, and it predicts the real-value dihedral angles of a protein backbone. The SPINEX method improved three-state secondary structure prediction to 85% when it was trained with 2640 sequences from the PISCES dataset and tested on 16 protein chains (Bradley, Misura, & Baker, 2005) whose lengths are between 49 to 88 residues.

2.3 Summary

PSS prediction from the information of protein sequences is a fundamental problem in the research areas related to protein structure determinations, and it poses common challenges as follows:

- The enormity and high dimensionality of a protein sequence's space in which twenty types of amino acids are linked in various orders and numbers.
- The tradeoff of sequence homology. This means that if the pairwise sequence similarity of the datasets is more than 25%, i.e. over the twilight zone (Rost, 1999) as explained in Section 1.5, the PSS prediction results are biased toward

homologous sequences in experiments. Otherwise, by setting the pairwise sequence similarity threshold to less than the twilight zone for all sequences in a dataset, the prediction performance is decreased due to the collection of non-homologous sequences, although, it results in a more accurate estimation of the generalization performance (especially on sequence and structure types that are not represented in the training set).

- It has been shown that many sequences with sequence similarity below the twilight zone have similar structures as mentioned earlier.

In general, the performances of PSS prediction methods are varied in different aspects such as using the information of single residues or sequence segments, protein profiles, amino acid encoding schemes, secondary structure assignments, quality of 3D structures, lengths of sequences, the sizes and sequence similarity levels of datasets, hybrid techniques, and consensus schemes. For PSS prediction methods, the aforementioned criteria in the experimental setups make objective comparisons of the performances difficult or in some cases impossible.

In Table 2.1, a number of PSS prediction methods whose experimental setups are different than those of the proposed PSS prediction model in this study and among themselves are listed with the brief descriptions of their differences and mismatches in experimental setups. PSS prediction has been greatly improved with the use of ML techniques, and therefore, it is justifiable to adopt a hybrid approach to aggregate the strengths of different techniques for improving PSS prediction accuracy. However, there is not a PSS prediction method that performs uniquely well for all types of secondary structures and quality of predicted secondary structures with different datasets.

Lastly, in this chapter, we explained various types of PSS prediction methods that have been developed over several years, and examined the methods' performances.

A number of approaches of protein representations and evaluations were discussed which are the essential introductions to Chapters 4 and 5 in which our proposed PSS prediction model, new amino acid encoding schemes and their corresponding experiments and performances are discussed. In the next chapter, we provide an overview of the principles of a number of important learning methods and protein structures which are the main components of the PSS prediction methods discussed in this chapter and the proposed PSS prediction model in Chapter 5.

Table 2.1: Examples of PSS methods whose results were exempted for performance comparison due to different experimental setups as explained in Section 2.3.

#	Outlines of different experimental setups in PSS prediction methods
1	Training set: cullPDB (6128 seq.), test set: (272 seq.), 30% sequence similarity, 8 classes (Zhou & Troyanskaya, 2014).
2	Training set: (ASTRAL40-1.73), test set: (ASTRAL40-1.75), sequence similarity≤40% (Bettella, Rasinski, & Knapp, 2012).
3	Training set: cullPDB (5534 seq.), test sets: CB513, (CASP10/11 228 seq., unspecified sequence similarity), 8 classes, PSI-BLAST profiles (Li & Yu, 2016)
4	A consensus approach with 17 prediction methods (Wei, Thompson, & Floudas, 2012).
5	Training set: (Chou 244 seq.), test set: (Chou 201 seq.), sequence similarity≥35%, 8-state (Chen, Chen, Zou, & Cai, 2009).
6	Training set: (undefined:knowledge based training), test set: (4 sequences) (Bingru, Wei, Zhun, & Huabin, 2009).
7	Training set: (dunbrack 1763 seq.), 3-fold cross-validation, sequence similarity≥30% (Madera, Calmus, Thiltgen, Karplus, & Gough, 2010).
8	Training set :(6048 seq.), 5-fold cross-validation, sequence similarity≥30% (Leman, Mueller, Karakas, Woetzel, & Meiler, 2013).
9	Training set: (cullPDB), test sets (RS 126 seq., CB 513 seq.), 8-state, sequence similarity≥30% (Wang, Zhao, Peng, & Xu, 2011).
10	Training set:(PSIPRED), training set: (CASP8), unspecified sequence similarity (Babaei, Geranmayeh, & Seyyedsalehi, 2010).
11	Training set: (cullPDB 5600 seq.), test sets: CB513, (CASP10/11 228 seq.), unspecified sequence similarity, PSI-BLAST profiles (Wang, Peng, Ma, & Xu, 2016).
12	Training set: cullPDB (7522 seq.), test set (1630 seq.), sequence similarity≥25%. Dividing the dataset to low/high resolution, no low resolution set in training, PSI-BLAST profiles (Mirabello & Pollastri, 2013).
13	A consensus approach with 100 BRNN, using structural similarity (Magnan & Baldi, 2014).
14	Traning set: (EVA 2713 seq.), test set : (125 seq.), unspecified sequence similarity (Green, Korenberg, & Aboul-Magd, 2009).
15	A consensus approach with 4 structure prediction methods (Qu, Sui, Yang, & Qian, 2011).
16	Training set: (1425 seq.), test sets: (CASP9/10 195 seq.), unspecified sequence similarity (Spencer, Eickholt, & Cheng, 2015).

Chapter 3

Background

3.1 Introduction

In the previous chapter, we introduced the problem of protein secondary structure prediction and a number of techniques that have been developed for protein secondary structure prediction, especially state-of-the-art methods based on ML approaches. In this chapter, we will explore other ML approaches in order to improve performance further especially in the context of more appropriate test sets and metrics, and all the pieces that are required to do secondary structure prediction and evaluate it well. We review several machine learning (ML) techniques which are: artificial neural networks, support vector machines, genetic programming, and clustering.

The ML methods are employed in the proposed protein secondary structure prediction model or used independently in the experiments for evaluations and comparisons. The concept, structure, learning techniques and applications of the ML techniques are explained in some details. In addition, we discuss protein structures and some related subjects such as the different types, geometries and properties of secondary structures in a computational context, and the techniques that are used for secondary

structure assignments. Amino acid encoding is an important step in ML methods for protein secondary structure prediction. We explain a number of amino acid encoding schemes. Lastly, two common evaluation techniques which are used in secondary structure prediction methods are discussed.

3.2 Artificial Neural Networks

In this thesis, we employ Artificial Neural Networks (ANNs), in a number of experiments such as the evaluation of common amino acid encoding schemes (Section 5.2.2) and the proposed protein secondary structure (PSS) prediction technique (Sections 5.3, 5.4 and 5.5) which are explained in detail in Chapter 5. In addition, we employ ANNs to construct independent PSS prediction models in order to compare the performance of the proposed PSS prediction model with those of state-of-the-art PSS prediction methods that commonly employ cascaded ANNs. Therefore, in this section the fundamental aspects of ANNs which are related to designs and learning algorithms are discussed. In addition, we review a number of important neural network architectures and applications in different classes of problems.

ANNs are one of the important field of studies in computational intelligence which have been applied successfully to a number of scientific, financial and engineering problems such as optimizations, classifications, controls, pattern recognitions, and data mining. ANNs are mathematical models inspired by the biological nervous systems (McCulloch & Pitts, 1943).

By the mathematical definitions, the main components of ANNs are artificial neurons, connection weights and activation functions. The weights correspond to the biological synapses which are the means of inputs to a neuron. An activation function which computes the weighted sum of the inputs of an artificial neuron is the simplified

representation of a biological neuron's function which has nonlinear and dynamical properties.

In a simple form, the output O of an artificial neuron is computed by a step function f which is a hard-threshold activation function as follows:

$$net = \sum_{i=1}^{n} w_i \cdot x_i - \theta \tag{3.1}$$

$$O = f(net) = \begin{cases} 1 & net \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

where n is the number of inputs to the neuron. x_i, w_i are the neuron's ith input and connection weight respectively, and θ is a threshold. net is the sum of weighted inputs of neuron i which is marginalized by threshold θ. An early ANN model was a single layer of neurons which is called perceptron. Perceptrons cannot solve problems which have linearly-inseparable spaces. A common ANN architecture is called a Multilayer Perceptron (MLP) which has three types of interconected neuron layers which are input, hidden and output.

In MLPs, information flows in a forward direction, from the input layer to hidden layer and from the hidden layer to the output layer, and such a neural network architecture is called a feed-forward ANN. A feed-forward ANN that employs a step function as an activation unit is limited to binary function approximations. Most neural networks employ differentiable functions such as the Logistic function which is a commonly used activation function in feed-forward ANNs for continuous function approximations, and it is defined as follows:

$$O = f(net) = \frac{1}{(1 + e^{-net})} \tag{3.3}$$

An important process for the training of ANNs is the adjustments of connection weights which are achieved by a learning algorithm. In perceptrons, for a neuron k

at time t a simple rule is used for weight adjustments as follows:

$$w_i(t+1) = w_i(t) + \eta(d^k - o^k(t)) \cdot x_i \tag{3.4}$$

where x_i, w_i are the ith input and weight of neuron k. η is a learning rate which adjusts the learning speed. d^k, $o^k(t)$ are the target and actual outputs of neuron k respectively at time t. The common learning algorithm for feed-forward ANNs which employ the Logistic function is called Backpropagation (Rumelhart et al., 1986). The algorithm is based on the gradient decent method, and it computes the partial derivative of a neural network's error with respect to each connection weight (Werbos, 1990). The error function and weight adjustment are defined as follows:

$$E = \frac{1}{2}\sum(t_i - a_i)^2 \tag{3.5}$$

$$\Delta w_{ij} = -\eta \cdot \frac{\delta E}{\delta w_{ij}} \tag{3.6}$$

where w_{ij} is the weight between the neurons i, j, and η is a learning rate. a_i is the actual output of an output neuron i and t_i is a target value for neuron i. By the Backpropagation method, a network's error is propagated from output layer to hidden layer(s) and from hidden layer(s) to the input layer, and in each step weight adjustments are performed.

In order to compute the error of a feed-forward ANN, inputs are propagated from the input layer to the output layer. In a feed-forward ANN that employs the logistic function, the output of neuron i from layer $k+1$, $a_i(k+1)$, is computed as follows:

$$net_i = \sum_{j=1}^{n} w_{ij} \cdot a_j(k) - b_i \tag{3.7}$$

$$a_i(k+1) = f(net_i) = \frac{1}{(1+e^{-net_i})} \tag{3.8}$$

where b_i is the bias (activation threshold) of neuron i, and $a_j(k)$ is the output of neuron j from layer k. Neuron j is connected to neuron i, and w_{ij} is their connection weight. net_i is the sum of weighted inputs of neuron i marginalized by bias b_i.

After the outputs of a feed-forward ANN are calculated, the errors are propagated to the network in backward order, and in each layer weight and bias modifications at time $t+1$ are calculated as follows:

$$\Delta w_{ij}(t) = -\eta \cdot \delta_i \cdot a_j + \alpha \cdot \Delta w_{ij}(t-1) \tag{3.9}$$

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) \tag{3.10}$$

$$b_i(t+1) = b_i(t) - \eta \cdot \delta_i \tag{3.11}$$

where w_{ij} is the connection weight from neuron j to neuron i and a_j is the output of neuron j. $\Delta w_{ij}(t)$ is the magnitude of the weight adjustment at time t, α is a momentum rate, and δ_i is the error value of neuron i. If neuron i is in an output layer, δ_i is calculated as follows:

$$\delta_i = a_i(1-a_i)(t_i - a_i) \tag{3.12}$$

whereas for neuron i in a hidden layer, δ_i is calculated as follows:

$$\delta_i = a_i(1-a_i)\sum_{k=1}^{n} \delta_k \cdot w_{ki} \tag{3.13}$$

where w_{ki} is the connection weight from neuron k to neuron i and n is the number of neurons in a hidden or input layer. To evaluate the prediction accuracy of an ANN, the root-mean square error (RMSE) of the outputs of the neural network is computed as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(t_i - o_i)^2}{n}} \tag{3.14}$$

where o_i, t_i are the actual and target outputs of neuron i, and n is the number of neuron in an output layer.

There are some variations of ANN architectures which have been developed for different applications, and these ANNs employ different nonlinear activation functions

and learning algorithms. Recurrent neural networks (RNNs) are one of the state-of-the-art ANN for nonlinear time series and temporal pattern classifications (Chambers & Mandic, 2001). In RNNs, the outputs are returned to the input layer.

The Self-organizing Map (SOM) is another ANN which can learn the distribution, categorization and topology of inputs (Hecht-Nielsen, 1989). A SOM is a grid of discrete points which are connected to input neurons. For each connection a weight is associated to measure the strength of the connection. A Radial Basis Function (RBF) network is a feed-forward ANN with three layers, and it uses a linear activation function for the output layer and a nonlinear function for the hidden layer. RBFs classify data by using hyperplanes, and unlike MLPs, RBFs are guaranteed to converge and learn faster.

Moreover, Deep Neural Networks (DNNs) are other variations of artificial neural network techniques based on Deep Learning whose robust performance has been evaluated in a number of important applications such as pattern recognitions (Cireşan & Meier, 2015). Deep learning methods learn feature hierarchies with features from higher levels of the hierarchy that are formed by the composition of lower level features. In other words, a deep learning model learns features without a prior knowledge at multiple levels of abstraction and enables a system to learn complex functions which map the input to the output directly from data (Bengio, 2009).

Lastly, there is not an ANN to be the best model for solving all problems, and each type of ANNs is a suitable solution for certain classes of problems. In a general term, it has been investigated that for static- and time-dependent optimization problems, the average performance of every two algorithms among all possible problems is identical which is known as No Free Lunch (NFL) theorem (Wolpert & Macready, 1997).

3.3 Support Vector Machines

In this study, we employ Support Vector Machines (SVMs), in a number of experiments such as the evaluation of the proposed codon encoding scheme in protein secondary structure prediction (Section 5.2.4). We compare the performances of our proposed Genetic Programming (GP) prediction model with those prediction methods based on one-tier SVMs (Section 5.3). In addition, we employ SVMs to construct an independent secondary structure prediction model in order to compare the performance of the proposed prediction model with those of state-of-the-art PSS prediction methods that commonly employ cascaded SVMs (Section 5.5). Thus, the fundamental aspects of SVMs which are the architectures and learning algorithms are discussed in this section. In addition, we review an important feature of SVMs which is the kernel technique.

SVMs are a set of related methods based on the principles of statistical learning theory. The pioneering study on SVMs was on hand-written digit recognitions for US postal services (Schölkopf, Burges, Vapnik, Uthurusamy et al., 1995). Thereafter, SVMs have been applied for a number of problems such as pattern recognition, text categorization, time series and bioinformatics. In general, SVMs are used for classification and regression problems.

In statistical learning theory, a learning machine chooses a function from a set of functions that minimizes a predefined risk known as empirical risk which depends on the training set and the complexity of the class of the function. However, it was shown that bound on the risk cannot be easily computed, and it does not measure the quality of an acquired solution(Vapnik & Chapelle, 2000). The limitations were resolved by introducing a concept to control the risk known as *margin*. In this way, the class of functions selected for the learning technique is a hyperplane with an assumption that an input space is separable as shown in Figure 3.1(a). A hyperplane

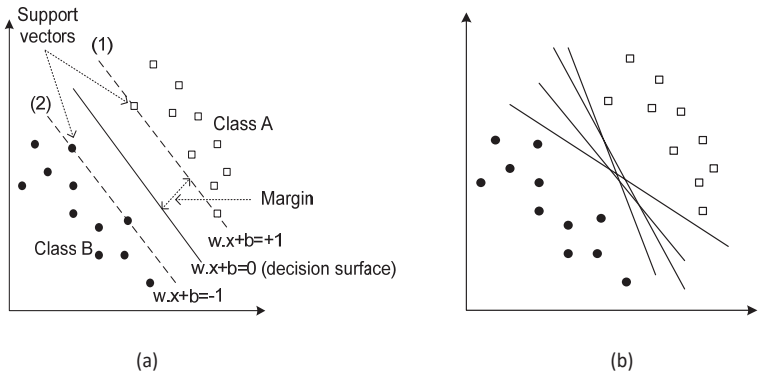

Figure 3.1: (a) A hyperplane which divides an input space into two separate classes. (b) Various hyperplanes which separate an input space.

that separates an input space is called the decision surface. In a problem, there are a number of hyperplanes that can separate the input space shown in Figure 3.1(b). The best hyperplane can be selected by imposing bound on the margin which is the minimum distance between a point in the input space and a decision surface. Thus, in an input space, the separation of input points is increased by choosing a wider margin.

In other words, a risk (error) is minimized by maximizing the margin's width. Therefore, the complexity of this class of functions is controlled by the margin. Moreover, the input points which have minimum distances from the decision surface are called *support vectors* which are on two hyperplanes that are parallel to the decision surface as shown in Figure 3.1(a). Therefore, SVMs can be formulated as a binary classifier and defined as follows:

$$f : X \to Y \quad , \quad X \in R^n \quad , \quad Y = \{-1, +1\} \tag{3.15}$$

By the definition, input points belonging to class A are assigned to $+1$, and those belonging to class B are assigned to -1. Therefore, the aim is to estimate function

f as follows:

$$f(x_i) = sign(w \cdot x_i + b) \quad , \quad i = 1, \cdots, n \quad , \quad x_i, w \in R^m, b \in R \tag{3.16}$$

where *sign* is the signum function. n, m are the number of input samples and the dimension of input space respectively. As shown in Figure 3.1, input point x_i on a decision surface satisfies $w \cdot x_i + b = 0$. Point x_i on hyperplane #1 or on the right side of the hyperplane satisfies $w \cdot x_i \geq +1$, and likewise, if point x_i is on hyperplane #2 or on the left side of the hyperplane, it satisfies $w \cdot x_i \leq -1$. Therefore, a constraint can be defined as follows:

$$y_i \cdot (w \cdot x_i + b) \geq +1 \quad , \quad y_i \in \{-1, +1\} \quad , \quad i = 1, \cdots, n \tag{3.17}$$

which satisfies all input points, except for the points between hyperplanes #1,#2. Meanwhile, the distance between two points x_i, x_j on hyperplanes #1 and #2 is equal to

$$\frac{|w \cdot x_i + b|}{||w||} + \frac{|w \cdot x_j + b|}{||w||} = \frac{|+1|}{||w||} + \frac{|-1|}{||w||} = \frac{2}{||w||} \quad , \quad i, j = 1, \cdots, n \tag{3.18}$$

where $\frac{2}{||w||}$ is the complete margin. Next, the aim is to find a hyperplane to maximize $\frac{2}{||w||}$ or to minimize

$$\min_{\{w,b\}} \quad \frac{1}{2}||w||^2 \tag{3.19}$$

subject to

$$y_i \cdot (w \cdot x_i + b) \geq +1 \quad , \quad i = 1, \cdots, n \tag{3.20}$$

The parameters of a hyperplane solution with the constraints is computed based on quadratic programming optimization technique using Lagrange function and Lagrange multipliers, $\alpha_i \geq 0$. The Lagrangian function of a given function $f(x, y)$ with constraint $g(x, y) = 0$ is defined as follows:

$$\partial f = \alpha \cdot \partial g \quad \text{or} \quad \partial L = 0 \quad \text{where} \quad L(x, y, \alpha) = f(x, y) - \alpha \cdot g(x, y) \tag{3.21}$$

Therefore, according to Equations (3.19), (3.20) and (3.21) Lagrangian function $L(w, b, \alpha)$ is defined as follows:

$$L(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i \cdot (y_i \cdot (w \cdot x_i + b) - 1) \tag{3.22}$$

The function L is at minimum where $\frac{\partial L}{\partial w} = 0$, $\frac{\partial L}{\partial b} = 0$, and based on Equation (3.22) the following equations are derived as follows:

$$\sum_{i}^{n} \alpha_i \cdot y_i = 0 \quad , \quad w = \sum_{i}^{n} \alpha_i \cdot x_i \cdot y_i \tag{3.23}$$

Also, according to Equation (3.21), $\alpha_i = 0$ for points that do not satisfy Equation (3.20) which means

$$\alpha_i \cdot (y_i \cdot (w \cdot x_i + b) - 1) = 0 \tag{3.24}$$

By replacing w from Equation (3.23) in Equation (3.22), function $L(w, b, \alpha)$ is redefined as follows:

$$\max_{\alpha} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \cdot \alpha_j \cdot y_i \cdot y_j \cdot (x_i \cdot x_j) \tag{3.25}$$

subject to

$$\sum_{i=1}^{n} \alpha_i \cdot y_i \quad , \quad \alpha \geq 0 \tag{3.26}$$

where function L is maximized based on the dual variable α_i. Using Equations (3.23) and (3.24), function $f(x) = \text{sign}(w \cdot x + b)$ and b are formulated as follows:

$$f(x) = \text{sign}(\sum_{i=1}^{n} \alpha_i \cdot y_i \cdot x_i \cdot x + b) \tag{3.27}$$

$$b = \frac{1}{|I|} \sum_{i \in I} (y_i - \sum_{j=1}^{n} \alpha_j \cdot y_j \cdot x_i \cdot x_j) \tag{3.28}$$

where I is a set of all points whose $\alpha_i \neq 0$.

In real world problems, input spaces are not always linearly separable as shown in Figure 3.2(a). Therefore, a solution based on hyperplanes may be practically impossible. A solution for a linearly inseparable space with n dimensions is to map data points to a higher dimensional space with m dimensions in which the transformed points are linearly separable as shown in Figure 3.2(b).

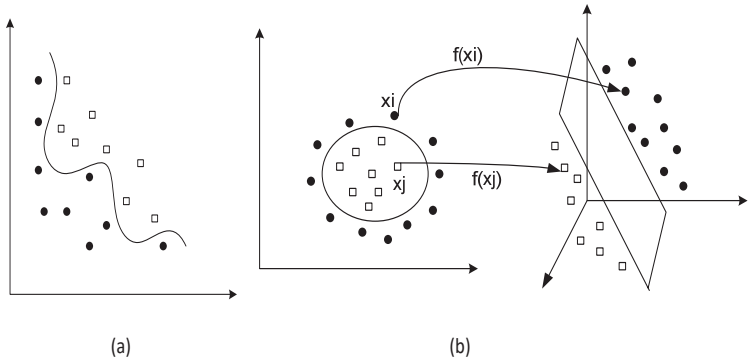

Figure 3.2: (a) A linearly inseparable space. (b) Mapping of a linearly inseparable 2D space to a linearly separable 3D space.

However, it is not easy to find a transform (mapping) function. According to Equation (3.27), the dot-product of every two points x_i, x_j in an input space needs to be computed, in addition of computing $f(x_i) \cdot f(x_j)$ where f is a transform function. The computation becomes even more costly when the number of input samples and the dimensions of original and transformed spaces are very high. For example, by using a simple transform function $f(x_1, x_2) = x_1^2 + x_2^2 + \sqrt{2}x_1 \cdot x_2$, 15 dot-products need to be computed for every two points.

A kernel function was introduced to overcome the increasing computation problem (Cortes & Vapnik, 1995). By the kernel function approach, for every two points x_i, x_j, the dot-product of $f(x_i) \cdot f(x_j)$ is not computed in a higher dimensional space. There are different types of kernel function such as linear function $k(x_i, x_j) = x_i^T \cdot x_j$

and sigmoidal function $k(x_i, x_j) = tanh(\gamma(x_i^T \cdot x_j) + \delta)$ where $\gamma > 0$.

In addition, the dot-product of $x_i \cdot x_j$ can be replaced by a kernel function in Equation (3.27) which means it is not needed to know directly a mapping function. A research area in SVMs is to know whether a kernel function corresponds to an exact dot-product of two points in a mapped space with a higher dimension.

An extension to SVM technique is soft margin SVMs which are less restrictive SVMs. Soft margin SVMs allow points that do not satisfy constraints defined in Equation (3.20) by using slack variables. A slack variable represents the distance of a point, located in the margin area of its opposite class label, from the hyperplane that includes the support vectors of the class label similar to the point's. By soft margin techniques Equations (3.19) and (3.20) are reformulated as follows:

$$\min_{\{w,b\}} \quad \frac{1}{2}||w||^2 + C\sum_{i=0}^{n}\varepsilon_i \tag{3.29}$$

subject to

$$y_i \cdot (w \cdot x_i + b) \geq 1 - \varepsilon_i \quad , \quad i = 1, \cdots, n \tag{3.30}$$

which minimizes the sum of errors. ε_i is an error corresponding to a slack variable and C is a cost coefficient to control an overfitting problem. There are two techniques for a multi-class classification with n labels by using SVMs, one-against-all and one-against-one which require n and $\frac{n(n-1)}{2}$ SVMs respectively, and then the classification results of SVMs are generalized by a voting scheme.

3.4 Genetic Programming

In this study, we propose a protein secondary structure (PSS) prediction model based on a Genetic Programming (GP) technique (Koza, 1992). We compare the performances of the proposed GP prediction model with those prediction methods based

on ANNs and SVMs (Section 5.3). In addition, we employ the GP technique in our proposed two-stage PSS prediction model (Sections 5.4 and 5.5). In order to provide a better overview of the experiments which will be described in Chapter 5, the fundamental aspects of GPs which are related to designs and learning algorithms are discussed in this section.

GP is an evolutionary computation technique which is an extensively studied field in Machine Learning (ML). GP is a heuristic search and optimization algorithm. The distinct features of GP make it a powerful technique for a range of classification and regression problems (Oltean & Dioşan, 2009).

The advantage of GP to other ML techniques is the flexibility that enables GP to be adaptable to the conditions of various problems. In GP, solutions are represented as variable length programs which do not require prior knowledge and logic as required by expert systems and artificial neural networks. The representation of a problem as a set of variable length programs instead of a specified length is the great advantage of GP, and the representation is considered as a superset of other possible ML representations (Koza, 1992).

The mechanism of GP is inspired from the theory of natural evolution, and the aim is to find a solution in the space of possible solutions. In general, GP applications are based on supervised learning by which training instances as inputs are associated to correct outputs where a fitness function compares the program's output with the desired result. A number of GP applications are also based on unsupervised training such as in reinforcement learning systems (Barto, Sutton, & Anderson, 1983) in which fitness functions have more roles than comparing the program outputs with desired targets.

In GP, a primary solution for a problem is represented as a tree. Initially, a population (generation) of random solutions (individuals) are created according to a

genotype scheme. To simulate the randomness of natural evolution pseudo-random numbers are used according to stochastic and probabilistic processes. In a generation, genetic operators transform the candidate solutions to new solutions. A new solution (individual) is passed to the next generation if the fitness of the new solution measured by a fitness function satisfies the conditions that are set for an individual to be a candidate solution.

A fitness evaluation is basically a mechanism which is provided to a learning algorithm, and it indicates whether an individual program is allowed to be multiplied and reproduced or be discarded form the population. In GP, unlike exhaustive and hill climbing searches, the search technique performed in a solution space is based on a beam search by which an evaluation mechanism (using a fitness function) chooses a set of possibly fitter solutions, and the rest of the solutions with lower fitnesses are discontinued.

In the search, the beam size is a population size which is equal to the number of individuals. Therefore, guided searches are performed more effectively by GP methods globally and locally for the exploration and exploitation of a solution space. In GP, there are three main types of genetic operators:

1. Crossover which swaps the selected genetic materials of two individual programs as shown in Figure 3.3.

2. Mutation which modifies the specific genetic materials of an individual program as shown in Figure 3.4.

3. Reproduction which creates the exact copy of a selected individual and places it into the population.

The structural units of individual programs in GP are terminals and functions. A function unit performs a defined operation on its inputs which can be terminals or

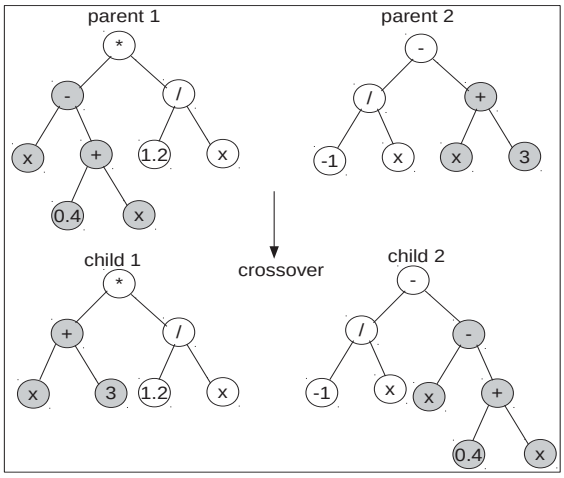

Figure 3.3: An example of a crossover operation on two individuals representing algebraic functions. The gray subbranches have been swapped between the two parents.

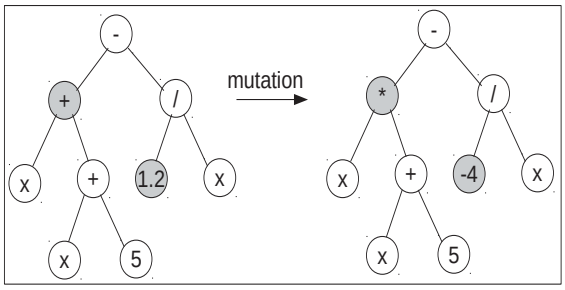

Figure 3.4: An example of a mutation operation on one individual. The gray nodes have been mutated.

the outputs of other functions. The set of terminals in a GP program are in fact the inputs of the program. In general, the choice of a function set depends on the domain of the given problem, and it can consist of arithmetic, boolean, relational, conditional, and transcendental functions.

An essential part in GP is a selection algorithm which is also called a selection operator. A selection operator is performed on the individuals of a population after their fitnesses are evaluated. The selection operator determines which individuals of a pop-

ulations should be selected, kept or replaced in the population. There are a number of selection methods such as fitness-proportional and tournament selection (Holland, 1992; Goldberg & Deb, 1991).

In Fitness-proportional selection, commonly used in genetic algorithms (GAs) (Holland, 1992), a probability p_i is assigned to individual i of a population with size n for reproduction based on its relative fitness f_i in the population as follows:

$$p_i = \frac{f_i}{\sum_{j=1}^{n} f_j} \tag{3.31}$$

The common selection algorithm in GP is tournament selection. The domain of competition in tournament selection is a subset of the population. In other words, each time a predefined number of individuals are randomly selected from the population and then the individuals compete and those with better fitnesses are chosen for reproduction. The number of individuals selected for tournament defines the tournament size. Selection pressure is high if a large tournament size is chosen, and selection pressure is low with a small tournament size.

Unlike the fitness-proportional technique, a centralized fitness comparison is not required between all individuals in tournament selection, therefore the selection technique is faster and speeds up evolution. The overall run of a GP algorithm is outlined as follows:

1. Set the numeric parameters of the GP program: population size, maximum individual size, crossover probability, selection method, and termination conditions.

2. Define functions and terminals.

3. Define fitness function.

4. Choose a selection method.

5. Initialize the population, randomly generated primary solutions.

6. Calculate a numeric fitness or rating value for every individual.

7. Choose individual/s from the current population based on the selection method.

8. Apply genetic operators on the selected individual/s.

9. Place the result of genetic operations into the new population.

10. Repeat from step 7 if the number of individuals in the new population is less than the predefined population size.

11. Replace the current population with the new population, and go to step 6 if the termination conditions (generalization error, maximum generation, maximum run,...) are not met.

12. Choose the best individual, identified by the best fitness in the population, as the final solution.

3.5 Clustering techniques

In this study, we propose an encoding scheme for representing amino acid sequences and protein profiles based on clustering and statistical information. The new encoding scheme is employed in the experiments that are illustrated in Sections 5.3, 5.4 and 5.5. In this section, we overview a number of prominent clustering techniques which are used in unsupervised learning, and aim to provide information that is more descriptive to the related experiments performed in Chapter 5.

Supervised and unsupervised learning are two main learning techniques in Machine Learning. In supervised learning, data samples are represented as n pairs of (X_i, y_i)

as follows:

$$X_i = (x_{i1}, x_{i2}, \cdots x_{im}) \quad , \quad i = 1, \cdots n \quad , \quad j = 1, \cdots m \tag{3.32}$$

where sample X_i is represented as an m-dimensional vector and element x_{ij} is called an attribute whose value can be categorical or continuous. Sample X_i is associated to y_i which is a known class label in the form of a nominal symbol or a numeric value. Therefore, the aim of a learning algorithm is to map data samples to the correct class labels.

However, in an unsupervised learning method, the class labels of data are not known, and the method explores and organizes (categorizes) data samples through the relations that exist among the attributes of data samples. Clustering is a common technique for unsupervised learning, and there are a number of clustering techniques such as hierarchical clustering (Guha, Rastogi, & Shim, 2001), error minimization clustering (Hartigan & Wong, 1979) and graph-theoretic clustering (Zahn, 1971). In hierarchical clustering, data are grouped by using two techniques:

1. Agglomerative (bottom-top).

2. Divisive (top-down).

In agglomerative technique, each sample X_i is initially assigned to one cluster. Then, the clusters are merged by computing the distances of every two clusters. The two clusters which have minimum distance (the least dissimilarity) among all other cluster pairs are merged. The distance of two clusters is computed in two ways, single-link and complete-link, as follows:

$$d(c_i, c_j) = min \quad \{d(x, y) | x \in c_i, y \in c_j\} \tag{3.33}$$

$$d(c_i, c_j) = max \quad \{d(x, y) | x \in c_i, y \in c_j\} \tag{3.34}$$

where i, j are the number of existing clusters in an iteration, and $d(x, y)$ is the distance between sample x of cluster c_i and sample y of cluster c_j. By single link method, the distance of two clusters is defined by the distance of the two closest samples from the two clusters. However, the distance of two clusters by complete-link approach is defined by two farthest samples from the two clusters. Unlike the single-link method, the complete-link approach induces compact clusters.

In graph theoretic clustering, data samples are represented as a complete graph G. Each sample is a node of graph G. The dissimilarity or distance of every two nodes is computed, and the distance is assigned as a weight to their edge. Next, a minimum spanning tree (MST) is constructed from graph G. Lastly, an edge with the largest weight, greater than threshold L, is identified and the edge is removed. Thus, two subgraphs (clusters) are formed. Removing the longest edges from subgraphs is repeated recursively until there is not an edge with a weight greater than L for all sub-graphs.

In error minimization clustering, data samples are initially partitioned into several clusters which are defined randomly or based on a heuristic technique. Next, the total error of clusters are computed, and then the error is minimized by relocating the samples among the clusters based on an iterative process. The sum of square error (SSE) is a common technique for error minimization. The SSE error of k clusters over n samples (points) is computed as follows:

$$SSE = \sum_{q=1}^{k} \sum_{i=1}^{n} \sum_{j=1}^{m} (x_{ij} - c_{qj})^2 \tag{3.35}$$

where x_{ij} is the jth element (attribute) of sample X_i which is an m-dimensional vector. Also, c_{qj} is the average value of the jth attribute of the elements in cluster q. The k-means clustering technique is based on error minimization clustering. For n samples (points), k-means clustering is performed as the following steps:

1. Define the number of clusters (k) and termination criteria, *SSE* threshold and maximum number of iterations.

2. Select randomly k points which are assigned initially to the centers of the k clusters. The k cluster centers are denoted μ_1, μ_2,..., μ_k.

3. For each point X_i in cluster p if the distance between X_i and cluster center μ_p, $d(X_i, \mu_p)$, is greater than threshold l, X_i is relocated to cluster q where $d(X_i, \mu_q) \leq l$ and $p, q = 1, \cdots, k$.

4. Repeat step 3 for all points located in the k clusters.

5. Compute the means of the attributes of all the points located in the k clusters and assign them to μ_1, μ_2, ..., and μ_k respectively.

6. Compute *SSE* based on Equation(3.35).

7. If termination criteria (minimum generalization error, maximum iteration,...) are not met, repeat the clustering process from step 3.

The time complexity of k-means clustering with n samples is linear, $O(n)$, whereas the time complexity of most hierarchical clustering methods is quadratic, $O(n^2)$. In hierarchical clustering, the number of clusters is defined during the algorithm, and the cluster results are the same in different runs. However, in k-means clustering, the number of clusters is determined randomly or based on a heuristic technique at the beginning, and k-means clustering may result different clusters in each run.

3.6 Protein structures

A protein is formed by a specific number and order of the twenty amino acids (AAs) which are linked by chemical bonds (Burkowski, 2008). In general, a protein structure

is categorized in four levels as follows:

1. Primary sequence— a sequence of amino acids which is defined by the alphabetical symbols of the twenty amino acids.

2. Secondary structure— a linear chain of common structural motifs which is defined by three structural patterns called α-helix (H), β-strand (E) and coil (C) as shown in Figure 3.5

3. Tertiary structure— a polypeptide chain is represented by the atomic coordinates of backbone and side chains in a 3D space as shown in Figures 3.7 and 3.6.

4. Quaternary structure— two or more interacting polypeptide chains (macromolecule) are represented by all atoms in a three-dimensional space.

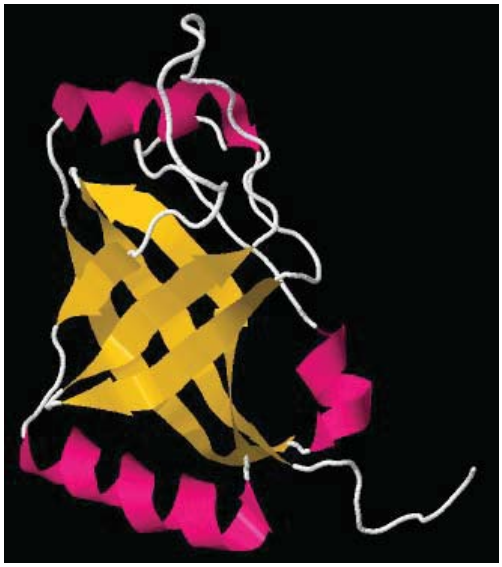

Figure 3.5: The structure of protein 1K8H represented by secondary structure elements. The secondary structures helices (H), strands (E), and coils (C) are shown in three different colours (yellow (E), red (H) and gray (C)).

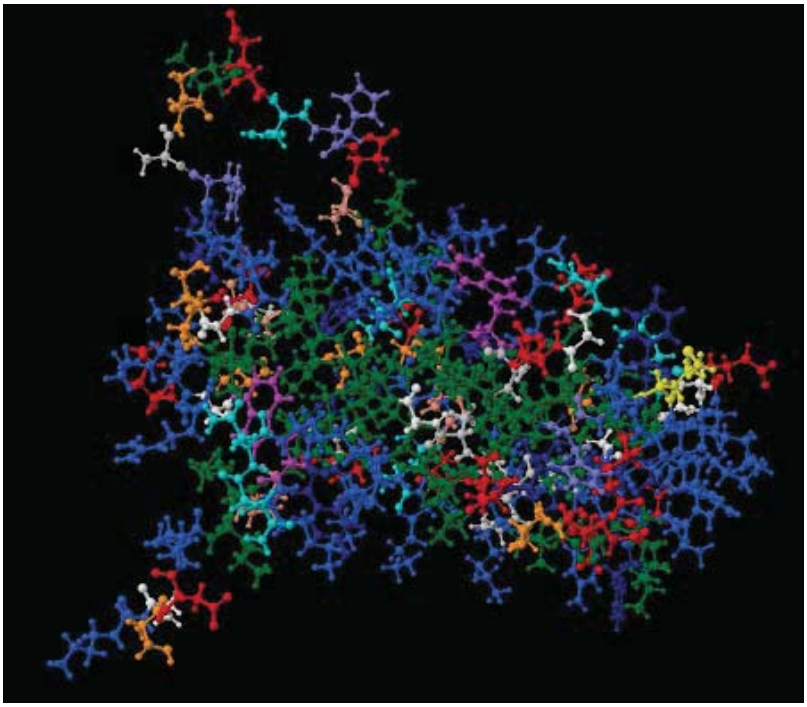

Figure 3.6: The 3D representation of protein 1K8H. Each type of the twenty amino acids is shown in a specific colour (Sussman et al., 1998).

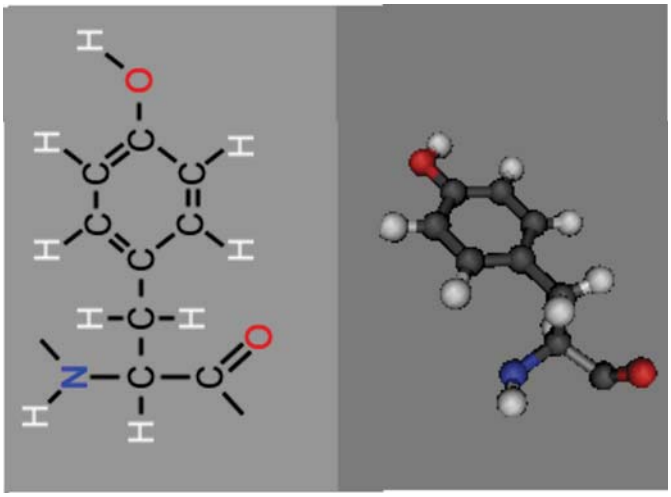

Figure 3.7: The skeletal formula and three-dimensional structure of the amino acid Tyrosin. Each type of atoms is shown in a specific colour.

The tertiary structure of a protein with fewer details can be defined by the dihedral angles of the protein's backbone atoms. A protein's side-chain atoms are not included in the representation of the protein's structure by dihedral angles. A protein's backbone consists of a number of peptide bonds which are sequentially linked and composed of nitrogen (N), hydrogen (H), carbon (C) and oxygen (O) atoms.

A peptide bond is formed by chemical reactions between the *carboxyl* group (-COOH) and *amino* group (-NH_2) of two adjacent amino acids as shown in Figure 3.8. Amino acids that are linked by peptide bonds are called *residues*. In order to represent

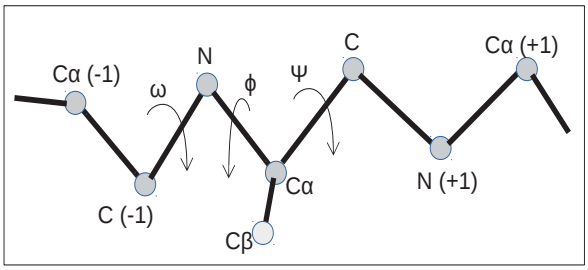

Figure 3.8: (a) A schematic view of a protein's backbone and the atomic orders. An amino acid's side-chain atoms are centered at C_β atom. A peptide bond is formed where dihedral angle ω is formed.

the conformation of a protein's backbone independently from the 3D coordinate system, three dihedral angles are computed for each residue. For two adjacent residues at the locations i, i+1 of a protein sequence with n residues, the dihedral angles of the residue r_i are defined by the order of four atoms as follows:

$$\phi: \ (N^{r_i}, \ C_\alpha^{r_i}, \ C^{r_i}, \ N^{r_{i+1}}) \tag{3.36}$$

$$\omega: \ (C_\alpha^{r_i}, \ C^{r_i}, \ N^{r_{i+1}}, \ C_\alpha^{r_{i+1}}) \tag{3.37}$$

$$\psi: \ (C^{r_i}, \ N^{r_{i+1}}, \ C_\alpha^{r_{i+1}}, \ C^{r_{i+1}}) \tag{3.38}$$

where $C_\alpha^{r_i}$ is the α-carbon atom of residue r_i to which a functional group (side-chain) is

linked. A side-chain which has specific chemical and structural properties determines uniquely the identity of an amino acid. Unlike the dihedral angle ω which is planar, approximately 180 degrees, the two dihedral angles ϕ, ψ are observed having various values from -180° to $+180^\circ$.

The degrees of $\phi - \psi$ dihedral angles are not randomly distributed from -180° to $+180^\circ$. In fact, the degrees of $\phi - \psi$ angles are in certain ranges due to steric constraints that exist among the atoms of the backbone and side chain. The distribution of secondary structures and energetically preferred ratios of $\phi - \psi$ angles can be illustrated effectively by the Ramachandran map (Ramachandran & Sasisekharan, 1968) in a two-dimensional plot ($[\phi, \psi]$ plot) ranging from -180° to $+180^\circ$ as shown in Figure 3.9.

In a $\phi - \psi$ plot, the distributions of ϕ, ψ dihedral angles in some regions are less than those of the other regions due to shorter *Van der Waals* radiuses, and even there is not any distribution of the dihedral angles in the rest of plot regions which are sterically disallowed. Moreover, the core (dense) regions are separated from each other to some degrees, and each core region corresponds to different types of secondary structures.

Moreover, the dihedral angles ϕ, ψ are sufficient to determine the three-dimensional shape of a protein's backbone with the assumption that the bond lengths and bond angles of the atoms do not change and remain fairly constant.

The formation of protein secondary structures had been speculated as certain local conformational patterns by Pauling and Corey (Pauling et al., 1951; Pauling & Corey, 1951a) several decades ago before the structures of Hemoglobin were experimentally determined. Protein secondary structures are the most common shapes that the segments of a protein chain adapt according to the hydrogen bonds detected among the residues of the segments.

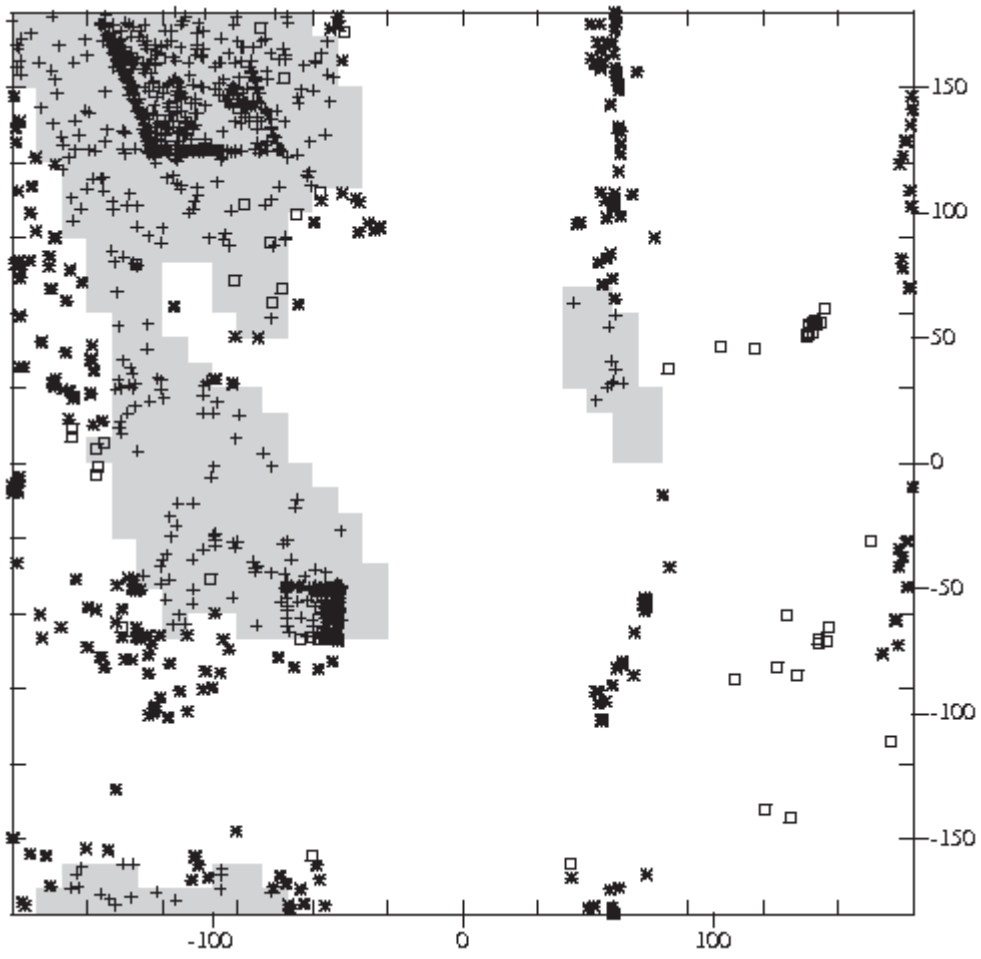

Figure 3.9: The Ramachandran map ($[\phi - \psi]$ plot) of protein 1K8H generated by Ramachandran server (Kleywegt & Jones, 1996). Plus and asterisk signs denoted for residues in core regions and outliers respectively.

In a peptide chain, the carboxyl oxygen of a residue forms a hydrogen bond with the amino hydrogen of another residue which can be at a close or far distance from each other along the chain. A helix is one type of protein secondary structure which is formed by a repetitive elementary structure and called *turn*. There are three types of helices as follows (Burkowski, 2008):

- α-helix is the most common helical structure in which each C_α atom has approximately 1.5Å distance from the helix axis. Each residue rotates 100° compared to the previous adjacent residue. Thus, a single turn in a α-helix contains 3.6 residues. More precisely a residue at position i participates in a hydrogen bond with a residue at position i+4.

- 3_{10}-helix is a rare helix found in short segments or at the end of an α-helices. It has 3 residue per turn which means ith residue forms a hydrogen bond with i+3th residue.

- π-helix is observed very rarely and it has 4.4 residues per turn. In a π-helix, ith residue forms a hydrogen bond with $i + 5$th residue along a protein chain.

The second type of protein secondary structures are β-strands which are the stretched segments of a protein chain formed by consecutive *bridges*. A bridge is a hydrogen bond between two residues that can be distant from each other along a protein chain. Two β-strands formed by repeating parallel or anti-parallel bridges form a *ladder*, and connected ladders form a β-sheet.

In parallel bridges, the indexes of residues on both sides of each hydrogen bond increase in the same direction wheras in anti-parallel bridges the indexes increase in opposite directions. A β-strand is a fully extended (stretched) fragment of a protein chain which usually consists of 3 to 10 residues (Kabsch & Sander, 1983).

The third type of protein secondary structures are coils which are irregular structural patterns. In other words, there are no regular and periodic relations between the dihedral angles of neighbouring residues in a fragment of a protein's backbone. A coil structure is more exposed to the surface of a protein and connects the other two types of regular secondary structures.

There are a number of methods to assign secondary structures to a protein's structure when all atomic coordinates are known, such as STRIDE (Frishman & Argos, 2004), XTLSSTR (King & Johnson, 1999), P-SEA (Labesse, Colloc'h, Pothier, & Mornon, 1997), KAKSI (Martin, Letellier, Marin, Taly, De Brevern, & Gibrat, 2005b), SECSTR (Fodje & Al-Karadaghi, 2002) and DSSP (Kabsch & Sander, 1983).

Moreover, a standard technique for secondary structure assignments is the DSSP method since the assignment technique is used more due to providing annotations for the Protein Data Bank (PDB) (Berman, Westbrook, Feng, Gilliland, Bhat, Weissig, Shindyalov, & Bourne, 2000), its applications are in critical assessment of protein structure prediction(CASP) (Moult, Pedersen, Judson, & Fidelis, 2004) and the evaluation of protein structure prediction servers (EVA) (Koh, Eyrich, Marti-Renom, Przybylski, Madhusudhan, Eswar, Grana, Pazos, Valencia, Sali et al., 2003). The DSSP method identifies eight secondary structures as follows:

- α-helix (H)
- 3_{10}-helix (G)
- π-helix (I)
- β-strand (E)
- Isolated β-bridge (B)
- Turn (T)

- Bend (S)
- Coil or all other irregular patterns (−).

However, in order to reduce the complexity of protein secondary structure prediction methods, in practice, the the eight secondary structures are regrouped to three structural classes, helices (H), strands (E) and coils (C). A secondary structure reduction is performed based on their structural similarities. The three commonly used method to reduce the eight states of DSSP to the three states are as follows:

1. PHD (Rost et al., 1993): $\{H, G, I\} \rightarrow H \quad \{E\} \rightarrow E \quad \{B, T, S, \text{-}\} \rightarrow C$
2. JNET (Cuff et al., 2000): $\{H\} \rightarrow H \quad \{E,B\} \rightarrow E \quad \{G,I,T,S,\text{-}\} \rightarrow C$
3. GORV (Kloczkowski et al., 2002) : $\{H\} \rightarrow H \quad \{E\} \rightarrow E \quad \{G,I,B\,T,S,\text{-}\} \rightarrow C$

The use of different eight- to three-state reduction methods causes approximately 3% changes in protein secondary structure prediction accuracy for the same predictions. In addition, after using a reduction scheme in a prediction method, some further changes are applied on secondary structure segments as follows (Rost et al., 1993):

1. $\{B\text{-}\} \rightarrow \{EE\}$
2. $\{B\text{-}B\} \rightarrow \{CCC\}$

3.7 Amino acid encoding schemes

Amino acid (AA) encoding is a an important step in Machine Learning methods. A protein sequence is represented by an alphabet of twenty letters. Therefore, protein

sequences are converted from nominal symbols to numeric values by an encoding scheme in order to be processed by ML methods. An AA encoding scheme has a great impact on the performance of any method. The quality of an AA encoding scheme depends on the amount of preserved information from a protein sequences and the dimenstionality of the encoding scheme's space. AA encoding schemes are grouped in two types as follows:

- Direct encoding
- Indirect encoding

In a direct AA encoding scheme, each AA in a sequence is assigned to a multi-dimensional vector. For instance, in an orthogonal encoding which is one type of direct encoding schemes, an AA is represented by a 20-dimensional vector whose one element that corresponds to the AA is set to 1 and the rest are 0's (Swanson, 1984). The orthogonal encoding alternatively is represented additionally by more dimensions which correspond to spacers (C- or N-terminus), unknown AAs, etc (Qian & Sejnowski, 1988). The Codon encoding scheme (Zamani & Kremer, 2011) is also a direct AA encoding which is based on genetic codon mappings and uses less dimensions compared to the orthogonal encoding.

The real values of physical and structural features of amino acids can be also used for AA encoding such as volume, mass, hydrophobicity (K&D), surface area, secondary structure propensity. Alternatively, the vectors of encoded AAs can be represented with fewer dimensions by grouping AAs based on physicochemical and structural features such as hydrophilicity, hydrophobic, charge, polarity, secondary structure propensity and exchange (substitution) groups (Taylor, 1986; Wu, Whitson, McLarty, Ermongkonchai, Chang et al., 1992).

AA direct encoding schemes are suitable for preserving the arrangement of amino

acids in a sequence and consequently the sequence local information. On the other hand, AA indirect encoding schemes preserve the global features of AA sequences, and are suitable for methods using sequences with different lengths. A common indirect encoding is the n-gram hashing method by which the frequencies of all n-letter AAs are computed and assigned to the elements of a 20^n-dimensional vector (Wu et al., 1992).

For instance with 2-gram encoding, an AA sequence with any length is converted to a 400-dimensional vector. Each AA frequency is computed by counting all possible adjacent AA pairs in a sequence. In order to reduce the dimensions of vectors, AAs are initially grouped based on their physicochemical properties as explained earlier. Then, the n-gram encoding is applied for sequences represented by an alphabetic set which contains less number of letters (Wu & McLarty, 2000).

3.8 Evaluation of secondary structure prediction

An important part in protein secondary structure (PSS) methods is the evaluation of predicted secondary structures. A common performance measure is the Q_3 score, a three-state score, by which predicted and observed secondary structures are compared for each residue. The Q_3 score measures a prediction accuracy based on a single residue and regardless of the prediction results of other local residues.

A secondary structure match receives a 1 otherwise 0, and the total Q_3 score of a sequence with n residues is calculated as follows:

$$Q_3 = \frac{q_\alpha + q_\beta + q_c}{n} \times 100 \tag{3.39}$$

where q_α, q_β and q_c are the total number of residues correctly assigned to a given secondary structure for helices, strands and coils respectively. The quality of pre-

dicted labels (classes) in a binary classification are evaluated with sensitivity, specificity, positive or negative predictive values, and Mathew's correlation coefficient measures (Matthews, 1975).

Similarly, in protein secondary structure prediction, which is based on a three-state prediction, the prediction measures are computed for predicted secondary structures using a 3×3 confusion matrix (Rost et al., 1993). By the method, two types of Q_3 measure are used for each secondary structure as follows:

$$Q_i^{obs.} = \frac{A_{ii}}{b_i} \times 100 \quad , \quad Q_i^{pre.} = \frac{A_{ii}}{a_i} \times 100 \quad , \quad i = C, E, H \tag{3.40}$$

where A_{ii} is the total number of all residues observed in class i and predicted in class i, b_i is the total number of residues observed in class i, and a_i is the total number of residue predicted correctly or incorrectly in class i.

An entropy-based information measure is also used to combine all elements of a 3×3 accuracy table to a normalized value. The normalized value is equal to 1 if a prediction is completely correct (Rost et al., 1993). By Q_3 scores, the average quality of predicted secondary structures are measured, and the prediction results may not indicate the quality of predicted segments of secondary structures in some cases.

Also, Mathew's correlation coefficient values which are calculated for the three secondary structures are an accurate way of examining the quality of predicted secondary structures for each class, whereas q_α, q_β and q_c counts do not reflect the prediction quality of helices, strands and coils separately.

Moreover, a segment overlap (SOV) score is another method to measure secondary structure prediction. The SOV score measures the segments of predicted secondary structures (Rost, Sander, & Schneider, 1994). The SOV score captures the correlation of the secondary and 3D structures which is a better similarity evaluation of 3D structure at the secondary structure level. For a sequence with n residues, the first

proposed *SOV* score which is also denoted *SOV94* was formulated as follows:

$$SOV = \frac{1}{n} \sum_{s \in A} \left[\frac{minov(s_1, s_2) + \delta(s_1, s_2)}{maxov(s_1, s_2)} \times |s_1| \right] \tag{3.41}$$

All secondary structure segments in observed and predicted sequences which have an overlap, at least one residue position in the same secondary structure, are represented by set A as follows:

$$A = \{\forall s_1, s_2 : s_1 \cap s_2 \neq \emptyset\} \tag{3.42}$$

where s_1 and s_2 are two segments in observed and predicted sequences respectively. The weight $|s_1|$ is the length of segment s_1. *minov* is the length of an overlap between the segments s_1, s_2 in a common secondary structure whereas *maxov* is the total extent of the segments s_1 and s_2 which is equal to $|s_1 \cup s_2| - |s_1 \cap s_2|$. The parameter δ allows minor variations at the edges (ends) of two segments by adjusting the ratio to 1. The parameter δ is set to a number smaller than $min\{minov(s_1, s_2), \frac{|s_1|}{2}\}$. The ratio in Equation (3.41) determines the quality of the match between s_1 and s_2.

An improved segment overlap measure is denoted *SOV99* and the score is based on *SOV94* (Zemla et al., 1999). The *SOV99* score corrects the normalization of n and the definition of δ. The *SOV99* values can be compared with other prediction evaluation measures due to the fact that the new normalization procedure bounds the values within [0,1] which are easily computed in a percentage scale. In *SOV99*, the parameter δ is computed as follows:

$$min\ \{maxov(s_1, s_2) - minov(s_1, s_2), minov(s_1, s_2), \frac{|s_1|}{2}, \frac{|s_2|}{2}\} \tag{3.43}$$

In Equation (3.43), the new definition of the parameter δ is considered as an equal share between predicted and observed segments. The segment overlap score indicates the correctly predicted segments with regards to observed segments. The accuracy of correctly predicted segments with regards to all predicted segments denoted $SOV^{pred.}$

is computed by assigning s_1, s_2 to predicted and observed segments as described in Equations (3.41) and (3.42).

In this chapter, we explained a number of ML techniques and complementary materials which are either used independently for protein secondary structure prediction and performance comparisons in the experiments conducted in Chapter 5, or employed in the proposed secondary structure prediction model. The concepts, structures and learning algorithms of ML techniques such as ANNs, SVMs, Clustering and GP were explained in detail in order to illustrate the learning procedures of the ML techniques, and describe the evaluation of the prediction methods' performances. In addition, protein structures were discussed in different structural levels, including secondary structures. Also, we discussed about a number of amino acid encoding schemes which are important to transform protein sequences to processable information in computational biology.

Chapter 4

Methodology

4.1 Introduction

In this chapter, we illustrate the proposed two-stage secondary structure prediction model. The structure, components, data representations and prediction phases of the proposed model are explained in detail. Initially, the first stage of the proposed model which is related to secondary structure transition site prediction is explained. Then, the second stage in which protein secondary structure prediction is performed is described. The proposed protein secondary structure model is also depicted with example data in order to describe the flow of information and the prediction phases.

4.2 The proposed PSS prediction model

In this study, we develop an *ab initio* protein secondary structure (PSS) prediction method based on a hybrid machine learning (ML) technique. In an *ab initio* PSS prediction method, the structural information of homologous sequences with known

conformations are not directly incorporated by the method. An *ab initio* PSS prediction method is complementary to other approaches based on two important criteria as follows:

1. A limited number of proteins with known structures are available compared to the vast number of amino acid (AA) sequences with conserved sequence information.

2. A high degree of sequence similarity is required between target and template sequences in order to incorporate the information of structural templates with a level of certainty. However, the similarity of many protein sequences that are compared to template sequences is less than 25% or below the twilight zone in which the structural similarity of protein pairs is reduced significantly. The relation between the sequence similarity and structural identity of protein pairs is explained in detail in Chapter 1.

The proposed PSS prediction method consists of two prediction models which collaboratively perform a three-state secondary structure prediction for α-helix (H), β-strand (E) and coil (C) as shown in Figure 4.1. The prediction is performed for a target protein sequence in two stages:

1. The prediction of secondary structure transition sites.

2. The prediction of secondary structures.

Secondary structure transition sites are locations on a target sequence where secondary structure changes occur such as a transition from an α-helix to a coil or vice versa as shown in Figure 4.2. The second prediction model incorporates *score vectors* for a secondary structure prediction as shown in Figure 4.1. The score vectors

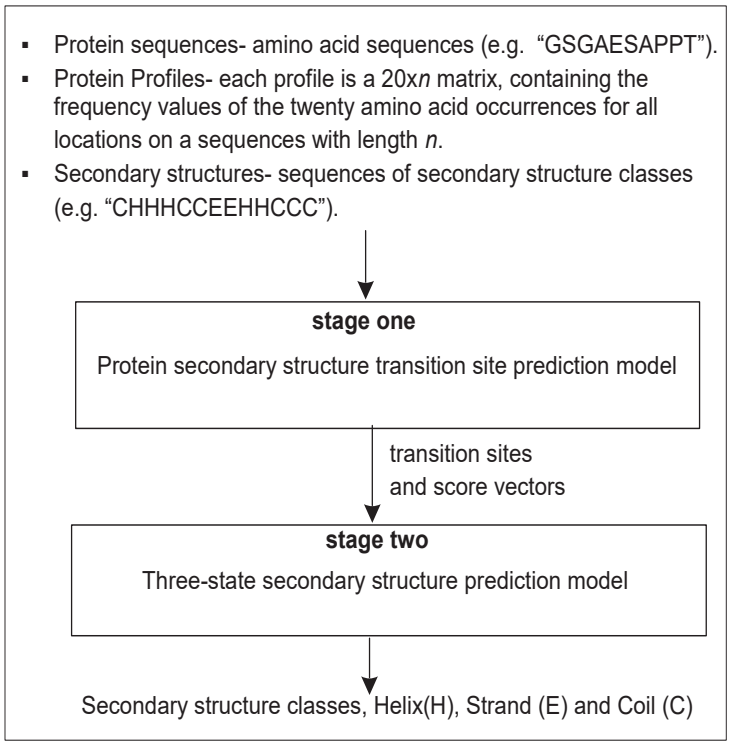

Figure 4.1: An overall view of the proposed prediction model. The secondary structure information which is used for computing the statistical information of clusters is derived from the 3D structures of the sequences in the training set. The detailed schemes of the two stages are provided in Figures 4.3 and 4.8.

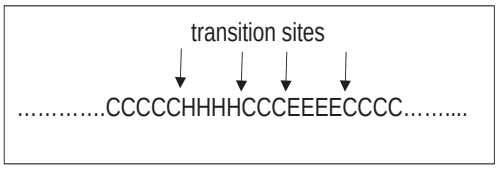

Figure 4.2: Protein secondary structure transition sites where secondary structure changes occur on a protein structure.

are generated by clustering, statistical information and PSS transition sites based on Ramachandran map and Information theory which are explained in details later on.

The secondary structure transition sites of a protein imply important information about the topology and long-range residue interactions of the protein. The prediction of secondary structure transition sites can be utilized for the homology modeling of protein structures when a segment of an amino acid (AA) sequence is defined by a homology tool to a secondary structure. However, the extent of the secondary structure on the either ends of the segment cannot be determined.

As explained in Section 3.6, the ϕ, ψ angles of residues in a protein's structure can be depicted by the Ramachandran map. The relation between residues and ϕ, ψ angles based on the properties of the Ramachandran map was examined by a number of protein structure determination methods to estimate dihedral angles.

However, in this study, the Ramachandran map is partitioned empirically to a predefined number of regions, and the statistical information of the regions and corresponding secondary structures are derived and incorporated in the proposed model for the prediction of transition sites and secondary structures. The relations between the regions of Ramachandran map and corresponding secondary structures are formulated by using statistical information extracted from protein sequences with known structures.

The transition site (TS) prediction model is a binary classifier. For the central residue of an AA segment, the TS model determines whether a secondary structure transition occurs at the position where the residue is located. As PSS prediction methods were reviewed in Chapter 2, the experimental results of the state-of-the-art PSS prediction methods indicate cascaded ML models perform better than one-tier ML models.

In this study, we employed a two-tier ML architecture in each stage of the proposed

PSS prediction model. The overall scheme of the TS prediction model in the first stage is shown in Figure 4.3. AA sequences are initially segmented by scanning a sliding window of length L along the sequences. For each AA segment, the profile information of L residues is mapped to a region on the 2D plot by using a feed-forward artificial neural network (ANN) in a *profile-region* tier. The number of input and output units and the information that the neural network in the *profile-region* tier incorporates are shown in Figure 4.4. A detailed view of the neural network learning in a *profile-region* tier is illustrated in Figure 4.5. In the next step, AA segments are encoded and clustered by using the Codon AA encoding scheme (Zamani & Kremer, 2012) and the k-means clustering technique (Hartigan & Wong, 1979).

Next, the statistical information extracted from clusters and identified regions are formulated to triplet *weight* scores. A triplet weight score is a three-dimensional vector, denoted by (w_1, w_2, w_3). The ith element of a triplet weight score, $w_i(s; g)$, is the information of pairing the secondary structure s of a residue in region g on the 2D plot. If central residue r in an AA segment is assigned to region g_k on 2D plot G and secondary structure s_i, weight score $w_i(s_i, g_k)$ is computed by Information theory as follows:

$$w_i(s_i; g_k) = \log_{10}\left(\frac{p(s_i|g_k)}{p(s_i)}\right) \ , \ s_i \in \{H, E, C\} \ , \ i = 1, \cdots, 3 \ , \ k = 1, \cdots, n \quad (4.1)$$

where $g_k \in G$ is defined by a feed-forward ANN in a *profile-region* tier, and n is the number of regions in G. $p(s_i|g_k)$ is the probability of secondary structure s_i given region g_k. In each cluster, the probabilities $p(s_i|g_k)$, $p(s_i)$ are computed from protein sequences with known structures.

Next, L triplet weight scores corresponding to an AA segment are presented to a feed-forward ANN in a *region-transition* tier. The outputs of the neural network determine whether the position of a central residue in the segment is assigned to a transition site. The type of information that the neural network in the *region-*

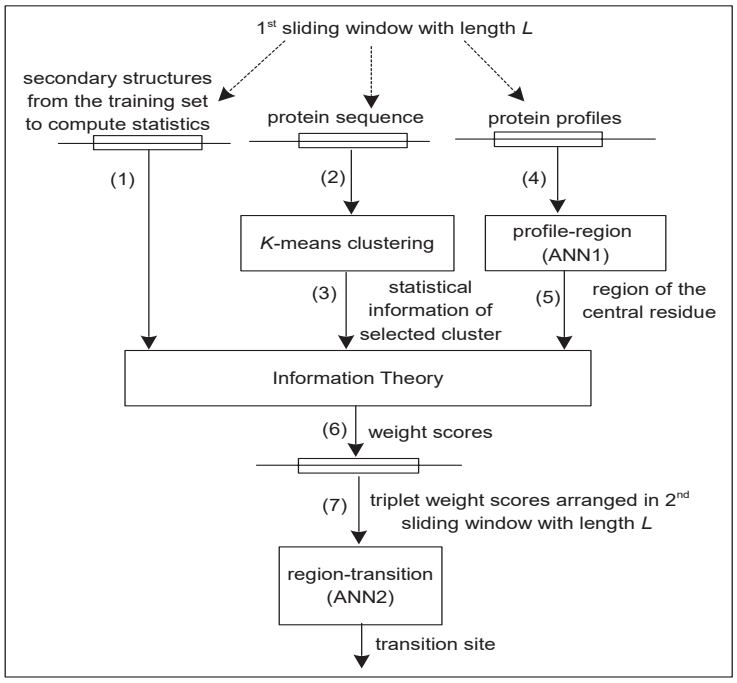

Figure 4.3: A detailed scheme of PSS transition site model in stage one. The "weight scores" are computed by using Equation (4.1).

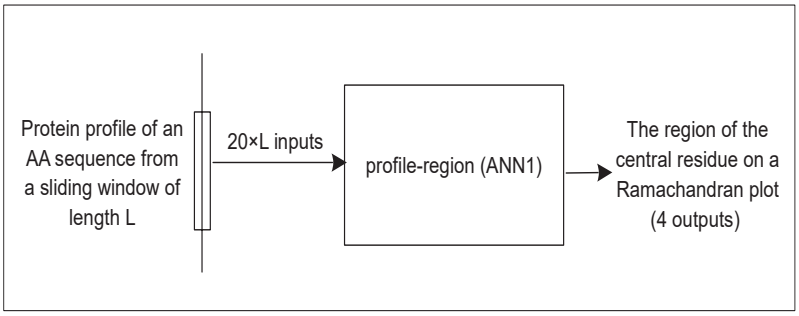

Figure 4.4: A detailed scheme of ANN1 in the *profile-region* tier. The number 20 represents the frequency values of the twenty types of amino acids that are occurred for each residue within a sequence segment that is defined by a sliding window of length L. Each output is a continuous value representing one of the equally divided regions based on dihedral angles (ϕ,ψ) on the Ramachandran map.

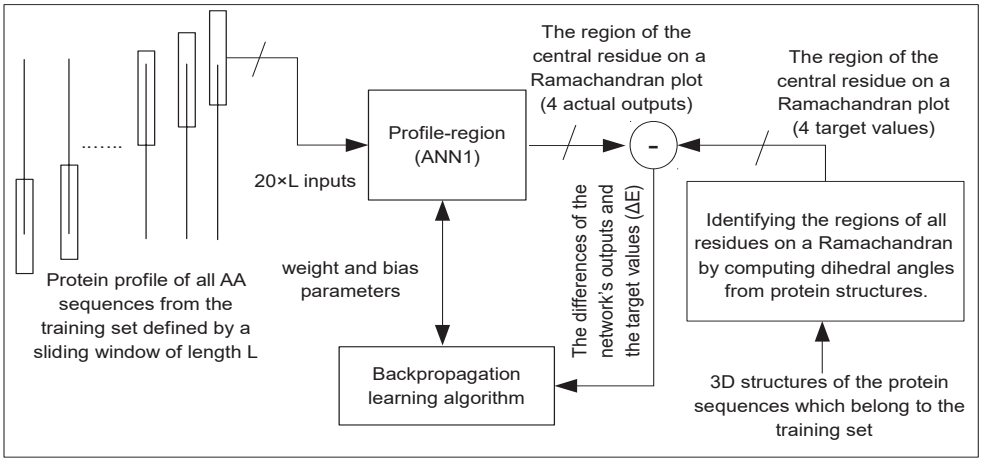

Figure 4.5: A detailed learning scheme of ANN1 in the *profile-region* tier by using a batch learning technique for all sequences of the training set in one iteration. The number 20 represents the frequency values of the twenty types of amino acids that are occurred for each residue within a sequence segment that is defined by a sliding window of length L. Each output is a continuous value representing one of the equally divided regions based on dihedral angles (ϕ,ψ) on the Ramachandran map.

transition tier incorporates, and the network's inputs and outputs units are shown in Figure 4.6. An overview of the neural network's learning detail in the *region-transition* tier is shown in Figure 4.7.

The inputs and outputs of the neural network in the *profile-region* tier are prepared as follows. For training sequences, the dihedral angles of the protein structures are calculated by using the atomic coordinates specified in the PDB (Sussman et al., 1998). The protein profiles are extracted from the HSSP dataset (Sander & Schneider, 1991). The secondary structure classes are defined by using the DSSP program (Kabsch & Sander, 1983) and assigned to the training sequences based on PHD scheme (Rost, 1996) as explained in Section 3.6. The Ramachandran map is divided into equal regions based on dihedral angles (ϕ,ψ), ranging from $-180°$ to $+180°$.

In the clustering component, all protein sequences are segmented by a sliding window of length L, and encoded by the Codon encoding scheme (Zamani & Kremer, 2012), and a k-means clustering is performed on the AA segments that are in a training set. For a target AA segment, the cluster to which the segment is mapped is initially identified. Then, the secondary structure information of all AA segments which are belonged to the identified cluster are passed to the Information theory component to compute the weight scores of the central residue of the target AA segment according to Equation (4.1).

The probability terms of Equation (4.1) related to the target AA segment are computed based on the statistical information that are extracted from the AA segments within the identified cluster. In stage two, a two-tier hybrid ML model is employed for secondary structure prediction as shown in Figure 4.8. For each residue, three types of inputs are incorporated in the second stage as follows:

1. Secondary structure probability.

2. Weight scores.

3. Transition sites.

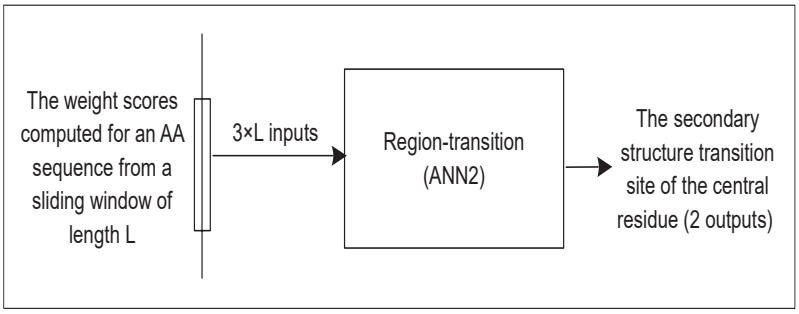

Figure 4.6: A detailed scheme of ANN2 in *region-transition* tier. The number 3 represents the three weight scores that are computed by using Equation (4.1) for each residue within a sequence segment that is defined by a sliding window of length L. The outputs are two continuous values which represent transition and non-transition sites.

We use the k-nearest-neighbor method to compute the probability of secondary structures. By the k-nearest-neighbor method, the AA segments of a cluster (templates) are compared with the segment of a target segment and scores are assigned to the template segments based on protein substitution matrices such as point accepted mutation (PAM) (Dayhoff, Schwartz, & Orcutt, 1978) and blocks substitution matrix (BLOSUM) (Henikoff & Henikoff, 1992) which are the standard benchmark tools to measure the protein sequence similarities. Next, the k segments with highest similarity scores are selected to compute the probabilities of the three secondary structures. The probabilities indicate the preference of secondary structures for the central residue of the target segment.

Theoretically, the three secondary structures are paired in nine ways such as EH, EC. Therefore, for two adjacent residues r_1, r_2 positioned in the middle of an AA segment which is mapped in region g, *score* vector A is constructed such that each

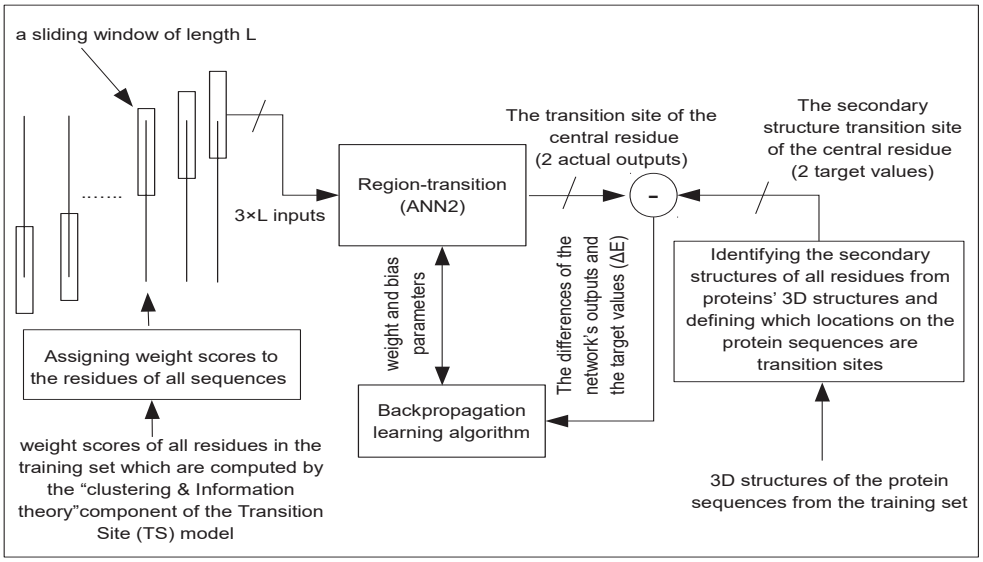

Figure 4.7: A detailed learning scheme of ANN2 in the *region-transition* tier by using a batch learning approach for all sequences of the training set in one iteration. The number 3 represents the three weight scores which are computed for the residues of all sequences in the training set by the "Clustering" and "Information Theory" components of the transition site (TS) model shown in Figure 4.2.

element of the vector a_i is computed as follows:

$$a_i = (w(s_1; g) + w(s_2; g)) \cdot p(s_1) \cdot p(s_2) \quad , \quad i = 1, \ldots, 9 \tag{4.2}$$

where s_1, s_2 are the secondary structures of residues r_1, r_2, and probabilities $p(s_1)$, $p(s_2)$ are calculated by the k-nearest-neighbor method. In addition, for each residue, we add the two outputs of the TS model to score vector A. It means a complete score vector consists of eleven elements. The outputs of the TS model are continuous values that indicate the likelihood of whether the central residue's location on a target AA segment is a transition or non-transition site.

In the second stage, L score vectors are presented to the input of ANN in the *transition-structure* tier where L is the length of a sliding window. The type of information that is incorporated to the neural network in the *transition-structure* tier, and the number of the neural network's input and output units are shown in Figure 4.9. The outputs of the feed-forward ANN in the *transition-structure* tier indicate the possible presences of secondary structures H, E and C at this intermediate step. Next, 3×L adjacent intermediate secondary structures are the inputs of the GP model in the *structure-structure* tier. The neural network's learning scheme in the *transition-structure* tier is illustrated in Figure 4.10.

A number of GP techniques have been developed for classification problems (Eggermont, Eiben, & van Hemert, 1999; Gathercole & Ross, 1994; Brameier & Banzhaf, 2001; Loveard & Ciesielski, 2001). The proposed GP method is a combination of two GP representation based on class enumeration and evidence accumulation schemes (Loveard & Ciesielski, 2001). A genotype is constructed in the form of nested "IF" rules as shown in Figure 4.11. In the GP method, the initial generation is populated with small individuals (simple solutions) which contain one or two nested "IF" rules that are initialized with random values. The tree representation of genotypes in GP resembles the structure of a decision tree such as C4.5 (Quinlan,

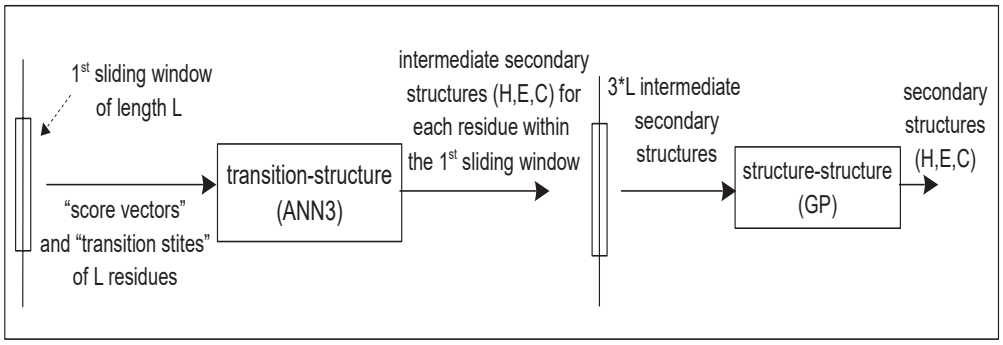

Figure 4.8: An schematic ML model for prediction of secondary structures in stage two.

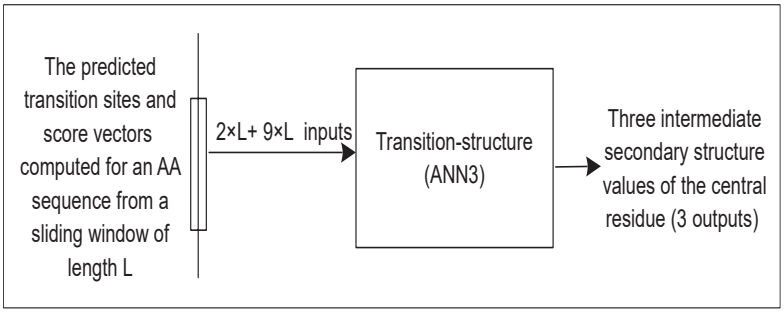

Figure 4.9: A detailed scheme of ANN3 in *transition-structure* tier. The numbers 2 and 9 represent the two predicted transition site values and the dimension of the "score vectors" whose elements are computed by using Equations (4.1) and (4.2) for each residue within an amino acid sequence segment that is defined by a sliding window of length L.

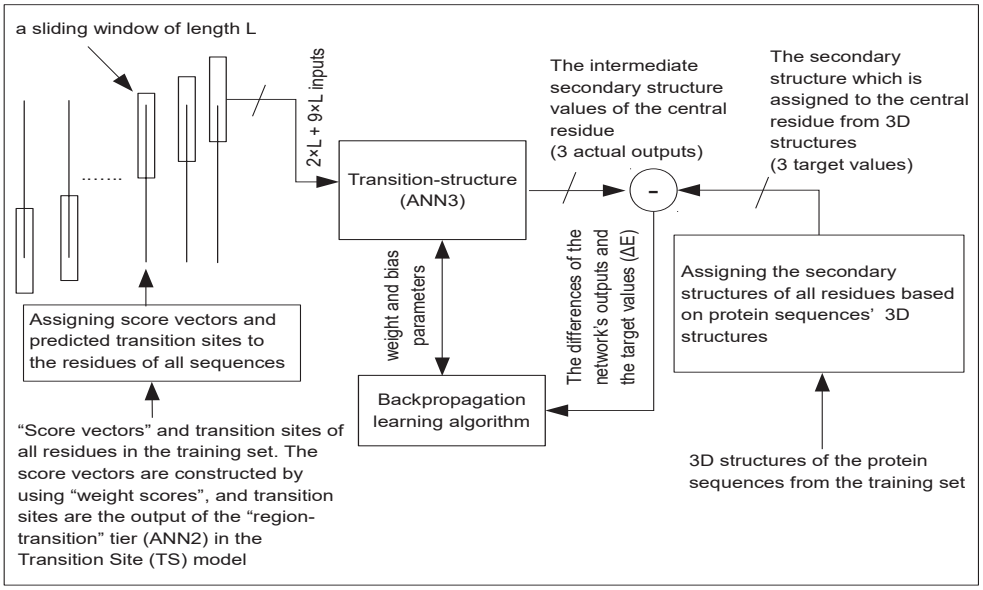

Figure 4.10: The ANN3's learning scheme in the *region-transition* tier by using a batch learning technique for all sequences of the training set in one iteration. The number 2 indicates the two predicted transition sites from the output of transition (TS) model as shown in Figure 4.2. The number 9 stands for the dimension of "score vectors" which are computed by using Equations (4.1) and (4.2) for the all residues of protein sequences in the training set.

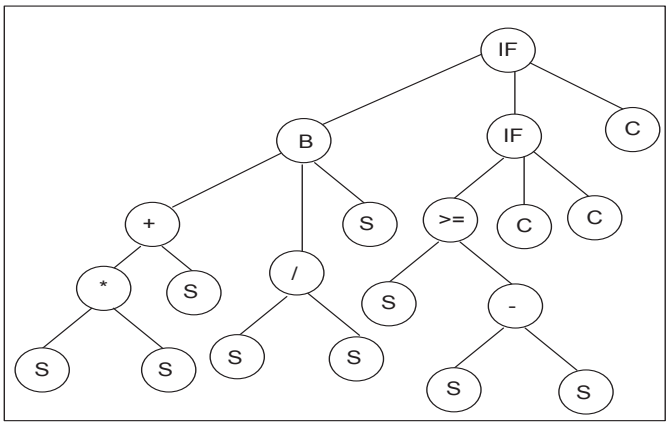

Figure 4.11: An example of a genotype representation in the proposed GP classifier.

2014). In GP, parent and leaf nodes are conventionally named *functions* and *terminals* respectively. The sets of functions and terminals applied in the proposed GP are shown in Table 4.1. A logical function returns True or False, and an arithmetic function returns a numeric value. The "B" function accepts three numeric values a, b, r and if $a \leq r \leq b$, it returns True, otherwise False.

The "C" terminal has a class label c and a weight w. The "S" terminal has two values l_1, l_2 which are the indices of locations on an AA segment with length L. For an AA segment (template), a tree evaluation starts from terminal nodes which are "S" and "C". A tree is traversed and evaluated based on a bottom-up approach which is explained in detail in Chapter 5. An "IF" function updates a certainty vector denoted

Table 4.1: Functions and Terminals

Node Type	Symbol
Arithmetic Function	$*, /, -, +$
Logical Function	$IF, B, >, <, >=, <=, =$
Terminal	C, S

cv, and the function has three arguments which are a logical function and two class terminals. The vector cv has three elements whose values are determined after a tree

evaluation is completed.

In an "IF" function, if the conditional argument is True, the ith element of the vector cv is updated, otherwise the jth element. The i, j values ($i, j = 1 \ldots 3$) are the class labels of the "C" terminals whose weight values are regarded to update the element of the vector cv. The conditional branch of an "IF" function is evaluated prior to the "C" terminals. If an "IF" function is a child node, the winning class's label is passed to the parent.

The overall evaluation of the genotype shown in Figure 4.11 is summarized for amino acid segment s in the following pseudocode section:

```
GenotypeEvaluation(node, s):
     if node is an amino acid segment terminal:
          - Calculate distance for s[i,j] and return the distance value.
     else if node is a class terminal:
          - Return class label and weight values of node[n,m].
     else if node is an arithmetic function:
          - Call GenotypeEvaluation with the node's left and right children.
          - Apply the node's arithmetic operation on the returned values of
            the node's children, and return the result.
     else if node is a logical function:
          - Call GenotypeEvaluation with the node's children.
          - Apply the node's logical operation on the returned values of
            the node's children, and return a True or False value.
     else if node is an IF function:
          - Call GenotypeEvaluation with the node's children.
          - Update the certainty vector based on the return values of
            the node's children by using the class label and weight values.
```

The genotype (individual) evaluation is performed by using the postfix traversal of the tree shown with example data in Figure 4.12. S[i,j] represents two locations i, j in an AA segment S defined by a sliding window of length L. C[n,m] represents class label n with a weight m. The three elements of vector cv which represent the three secondary structures are set to 0. For each S[i,j], the intermediate secondary structure values of AAs that are in location i, j are derived from the output of the first tier in secondary structure prediction model as shown in Figure 4.8. The output values for a specified location are real values from 0 to 1, indicating the likelihood of secondary structures H, E and C. For example, if A(0.2,0.45,0.6) and B(0.7,0.15,0.5) correspond to locations 1 and 9 respectively for an AA segment S, the euclidean distance of the two vectors A, B is equal to 0.59 which is the output of node S[1,9]. AA segments are defined by moving a sliding window which contains a predefined number of residues along protein sequences. Similarly, the rest of S[i,j] nodes' outputs are computed. A typical evaluation of the genotype shown in Figure 4.11 is illustrated with numeric values that are randomly assigned to the genotype and shown in Figure 4.12. The entire evaluation of the genotype is organized as the following steps:

1. s[1,9]= 0.59
2. s[3,5]= 0.3
3. 0.59*0.3= 0.177 ← Steps 1, 2
4. s[5,8]= 0.65
5. 0.177 + 0.65 = 0.827 ← Steps 3, 4
6. s[2,11]= 0.1
7. s[6,10] =0.4

8. $0.1/0.4 = 0.25 \leftarrow$ Steps 6, 7

9. s[4,8] = 0.6

10. $0.827 \notin [0.25, 0.6] \Longrightarrow B[0.827, 0.25, 0.6] = \text{FALSE} \leftarrow$ Steps 5, 8, 9

11. s[9,7]= 0.45

12. s[6,11]= 0.4

13. s[1,3]= 0.2

14. 0.4-0.2 = 0.2 $\leftarrow$ Steps 12, 13

15. $0.45 \geq 0.2 = \text{TRUE} \leftarrow$ Steps 11, 14

16. n=2 (class label), m=0.2 (weight)

17. n=3 (class label), m=0.4 (weight)

18. $cv(0, 0.2, 0) \leftarrow$ Steps 15, 16, 17

19. n=1 (class label), m=0.1 (weight)

20. $cv(0, 0.2, 0.1) \leftarrow$ Steps 10, 18, 19

21. $max(0.1, 0.2, 0) = 0.2$

In the example, if the order of secondary structure labels for certainty vector cv is helix (H), strand (E) and coil (C) respectively, the central residue of AA segment S is classified to secondary structure E after the genotype evaluation.

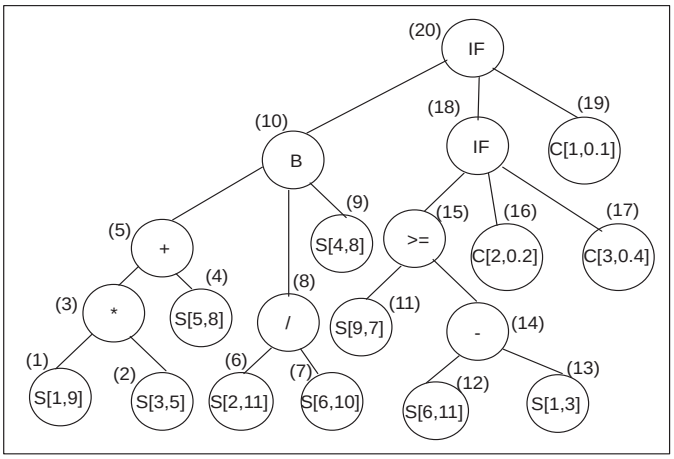

Figure 4.12: An example of a genotype representation with sample data in the proposed GP classifier. S[i,j] represents two locations i, j in an AA segment. C[n,m] represents class label n with weight m. The numeric labels of the nodes indicate the order of the tree evaluation.

4.3 Summary

In this chapter, we illustrated our proposed two-stage protein secondary structure prediction model. The two stages of the prediction model were described in detail which was complemented with schematic views of the model, equations, sample data representation and the evaluation of the genotypes in the proposed GP model. In this chapter, we provided the description of the proposed model and its components which are required for a series of experiments that are performed in Chapter 5.

Chapter 5

Experiments

5.1 Introduction

In this chapter, we illustrate the various experiments that were performed in this research. A number of amino acid (AA) encoding schemes are explained including the proposed AA encoding scheme denoted Codon. We discuss our new technique based on substitution matrices for the evaluation of AA encodings as a general approach in machine learning (ML). The codon AA encoding scheme is investigated further for PSS prediction by using various support vector machine (SVM) architectures, and the effectiveness of the proposed encoding scheme is compared to that of the commonly used orthogonal encoding scheme.

The next group of experiments that were conducted in this study are related to the proposed GP model and a new technique to encode protein sequences by using clustering and nearest-neighbor techniques. The efficiency of clustering and performance of the GP prediction model in PSS prediction are compared to those of two commonly used ML methods which are artificial neural networks (ANNs) and SVMs.

The third group of experiments are related to the proposed two-stage PSS prediction model. The last set of the experiments were conducted based on four criteria: (1) incorporating the information of PSS transition sites, (2) the sizes of protein datasets, (3) the performance of the PSS transit site (TS) model, (4) evaluation techniques and statistical analyses. The proposed two-stage model is initially evaluated without incorporating PSS transition sites with two common datasets that have small and medium sizes, approximately twenty four thousand residues and one hundred and twenty five thousand residues respectively, and we used the conventional approach to generalize the results.

Lastly, we used the latest dataset with a very large size, over two million residues, and recommended an evaluation approach based on a statistical analysis and sequence similarity for a comprehensive performance evaluation of PSS prediction methods.

5.2 Amino Acid Encoding

Recent technological advancements in high throughput sequencing have provided millions of DNA and protein sequences. As a result, an imbalance between data acquisition and complementary analytic techniques has been significantly increased. Machine Learning (ML) is a promising approach for extracting useful rules and patterns automatically from acquired data and performing more accurately and faster feature selections, classifications and predictions.

In ML techniques, an important issue is how to encode data in a format suitable for a computational tool. An encoding method has a profound effect on the applicability and resulting quality of a ML technique. An encoding scheme must preserve the information which is necessary to solve a problem such that inputs that should generate different outputs in the problem space must be distinguishable. Moreover, an

encoding scheme should attenuate noise in the input data and shield the subsequent processing from extraneous data. In an efficient encoding scheme, groups of input patterns which are supposed to generate similar output patterns are represented in a similar, or consolidated way.

By an encoding method, distinguishability and consolidation are problem specific concepts, which means an important piece of data required to solve one problem can be noise in another problem (Swanson, 1984; Maetschke, Towsey, & Bodén, 2005; Zimmermann & Gibrat, 2010). In bioinformatic problems, encoding input patterns is challenging where data instances vary in sizes from one sample to another and embody a structural or sequential component.

In this section, we focus on the encoding of protein sequences which are composed of chains of amino acids (AAs) of different lengths. The varying length nature of protein chains was addressed by using a sliding-window technique or a recursive network (Pollastri et al., 2002; Qian & Sejnowski, 1988; Rost et al., 1993). However, every AA within the sliding window or recursive network needed to be eventually encoded.

We evaluate various amino acid encodings based on their capability to approximate commonly used substitution matrices via best-trained ANNs. The substitution matrices such as point accepted mutation (PAM) (Dayhoff et al., 1978) and blocks substitution matrix (BLOSUM) (Henikoff & Henikoff, 1992) are the standard benchmark tools to measure the sequence similarities of protein sequences.

In this thesis, we propose a new AA encoding scheme, *codon* encoding (Zamani & Kremer, 2011), and the performance of the proposed encoding scheme was compared to a number of commonly used AA encoding schemes. The aim was to provide evidences to select confidently the best encoding method. In order to evaluate different AA encoding methods, we need to develop a benchmark test set. The primary sources

of protein information are AA sequences which can be used to predict a number of protein properties such as interactions, disulphide bonds, shape, secondary structure, etc.

There are many different possible test sets to choose from, but the aim is to select the most generic test set whose results would be applicable to the largest variety of ML problems. We selected a test set according to two assumptions: (1) the goal of any ML approach is ultimately to predict the function of a protein. Most ML methods that incorporate protein sequences achieve this goal by exploring some aspect of the problem such the shapes of proteins or their chemical properties, and (2) in related organisms, biological function tends to be preserved across related proteins, and the conservation of specific amino acids in the specific locales of a protein is a valid objective to predict in a comparison of encoding schemes.

Therefore, we decided to base the test set on the substitutability of specific amino acids in biological organisms and predict specifically similarity matrices which measure such substitutability.

5.2.1 Substitution Matrices as a Benchmark

Sequence alignments can reveal the underlying relations existing among protein sequences such as evolutionary distances, functions and structures. In general, sequence alignments are performed in two ways: (1) global alignment in which AA sequences are aligned with regards to the entire length of the sequences. The aligned sequences could be either originated from the same ancestor or different evolutionary paths and families, and (2) local alignment in which segments of protein sequences are searched for similarity and evolutionary information.

In sequence alignments, exploring the evaluation of sequence similarities is accomplished by a dynamic programming algorithm which attempts to find the most likely

alignment between the amino acids of two or multiple sequences (Needleman & Wunsch, 1970). The algorithm uses a probability measure based on similarity matrices and determines the likelihoods of substituting, deleting and inserting symbols in one sequence until it matches the other.

The mutation data matrices or point accepted mutation (PAM) matrices (Dayhoff et al., 1978) are one of the standard models of measuring sequence similarities based on likelihoods in sequence alignment methods. PAM matrices were extracted from an empirical dataset of 1572 mutations in 71 groups of closely related proteins and generated by measuring substitution frequencies in sequences aligned by human experts with 85% similarities or higher. PAM matrices were indexed with a numeric value such as PAM10 which means the substitution probabilities were derived from the sequences that have ten mutations in one hundred amino acids. Therefore, large indices imply more divergence among reference sequences.

An alternative to PAM is blocks substitution matrix (BLOSUM) (Henikoff & Henikoff, 1992) which can detect protein sequences with higher evolutionary distances. The BLOSUM matrix was generated as follows: (1) a frequency table was derived from a database of ungapped blocks by counting the relative frequencies of amino acids and corresponding substitution probabilities. Each block represents the conserved region of a protein family, (2) a logarithm of odds matrix for 210 possible substitutions based on the twenty amino acids is calculated. A log-odds score in BLOSUM represents the ratio of two amino acids' frequency of appearance in natural sequences to the two amino acids if appeared randomly based on their independent frequencies.

The numeric index of a BLOSUM matrix determines the level of similarity among reference sequences, and higher numbers indicate higher sequence similarity. In BLOSUM, the frequencies are calculated from amino acid blocks that are highly conserved regions, regardless of the evolutionary distances of referenced sequences, whereas mu-

tation frequencies in PAM are estimated from the sequences of closely related proteins.

5.2.2 Learning Substitution Probabilities

As explained earlier, our goal is to predict substitution matrices based on the premise that if an encoding scheme allows a ML technique to effectively predict the likelihoods of AA substitutions that occur in nature, then the same encoding scheme is expected to perform well on a number of different ML tasks involving the prediction of protein structure, composition, function, etc. The likelihoods in substitution matrices are numerical values as opposed to categorical. Thus, we required a ML regression method.

In order to keep things simple, we used a multilayer feed-forward ANNs that measured the effectiveness of each encoding scheme based on the speed of learning and the generalization error. The neural network consists of one hidden layer, and the output layer has one output neuron and the number of input layer's neurons is varied and depends on the selected encoding scheme. For each encoding scheme, the supervised training of the neural network is performed by gradient descent backpropagation learning algorithm (Rumelhart et al., 1986; Werbos, 1990).

In a training epoch, 210 pairs of encoded AAs are fed to the neural network after normalizing an input vector to $\{-1, +1\}$ The output errors are computed according to the corresponding substitution values of each amino acid pairs from substitution matrices. A training set is generated by applying each AA encoding on all 210 pairs of AAs. Next, a feed-forward ANN maps each sample of the training set to the corresponding entry of a chosen substitution matrix. The substitution matrices used in this study are BLOSUM50, BLOSUM62, BLOSUM80, PAM120 and PAM250. Therefore, five neural networks are trained for each AA encoding scheme.

The performance of each neural network is evaluated based on the average root

mean square error (*rmse*) in 10 runs since the training of a neural network results a different final *rmse* that is achieved by a different number of training epochs in each run. For each encoding scheme, the training is performed until a stop criterion is reached. In this experiment, a simple threshold error, as is convention in many ANN applications, is counter productive to define as a stop criterion since our goal is to evaluate the effectiveness of a variety of encoding schemes with different expected performances. Instead, we attempt to identify the convergence of the network which is defined as a very small reduction in error (or an increase in error) over 5 training epochs. The stopping criterion is defined specifically as follows:

$$E(t-5) - E(t) \leq \Delta E. \tag{5.1}$$

where E, t are the average *rmse* and epoch number respectively, and ΔE is set empirically to $10e-7$.

5.2.3 Evaluation of Encoding Schemes

Amino acid encoding is an important step to apply ML techniques in computational biology and aims to capture the underlying similarity and differences of amino acid pairs from symbolic data in a naturally meaningful way like that provided by substitution matrices. We evaluated fifteen amino acid encoding schemes as described in Table 5.1. Each amino acid encoding scheme generates n-dimensional vectors in Euclidean space that correspond to the twenty amino acids (AAs) (Swanson, 1984).

A number of the encoding schemes are applied commonly in the literature such as orthogonal (Baldi & Brunak, 2001), BLOMAP (Maetschke et al., 2005), polar distribution (Hu, Pan, Harrison, & Tai, 2004) and physicochemical properties (Lac & Kremer, 2009), based on the classification of AA conservation (Taylor, 1986). Except our new proposed encoding scheme #12, the other encoding schemes are based on

the grouping of various properties of AAs similar to the encoding schemes introduced in (Wu & McLarty, 2000), (Zvelebil, Barton, Taylor, & Sternberg, 1987).

In the AA encoding schemes, using a few combinations of AA features is worthwhile to examine the efficiency of them in ML applications. The properties range from AA preferences to form different types of secondary structures, molecular properties such as formula and molecular weight. A comprehensive list of amino acid features can be found in the AAindex database(Kawashima & Kanehisa, 2000) which contains around five hundred features.

In this study, we identified effective encoding schemes by using substitution matrices, as opposed to constructing an encoding scheme specifically designed to compute substitution scoring matrices (Zimmermann & Gibrat, 2010). Therefore, we aimed to provide a general assessment of the performance of AA encoding schemes across a broad class of problems, rather than a specific method to perform optimally on the scoring matrix problem.

In addition, we proposed a new AA encoding scheme called *codon encoding* (Zamani & Kremer, 2011) according to the genetic codon mapping method that is the building block of AAs. Each codon is the combination of three naturally selected nucleotides from the set {A,T,G,C} which create one of the twenty AAs. In addition, each AA can be derived from multiple codons.

For example, Valine is derived by one of the codons from the set {GTT, GTC, GTA, GTG}. The codons which form Valine can be represented by a directed graph as shown in Figure 5.1. The graph consists of four nodes which represent the nucleic acids {A,T,G,C}, and directed vertices for each successive pair of nucleic acids in one of the four Valine encodings G$\rightarrow$ T,T$\rightarrow$ T;G$\rightarrow$ T,T$\rightarrow$ C;G$\rightarrow$ T,T$\rightarrow$ A;G$\rightarrow$ T,T$\rightarrow$ G. Next, the graph is represented in a 4$\times$4 connectivity matrix which is converted to a 16-dimensional vector of 0's and 1's. Similarly, we constructed the graphs of the

other AAs by their corresponding sets of codons.

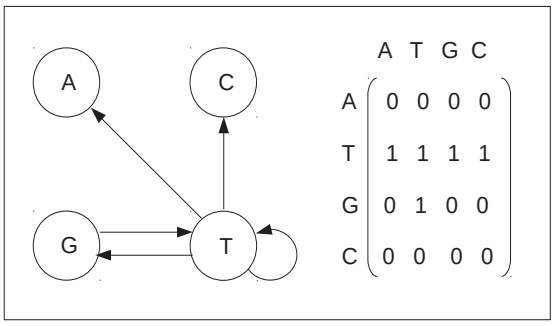

Figure 5.1: The graph and 4×4 connectivity matrix representations of the codons forming the amino acid Valine based on the four nucleotides {A,T,G,C}.

The experimental results indicate that the training accuracies of the encoding schemes #11, #12, #14 and #15 are better than those of the other encoding schemes as described in Table 5.1. The four best encoding schemes were identified based on the average *rmse* when the termination criterion defined in Equation (5.1) is met. The four best encoding schemes are compared based on the average *rmse* and the number of required epochs for training as shown in Figures 5.2 and 5.3 respectively.

The complete training results of all AA encoding schemes with BLOSUM62, BLOSUM80 and PAM250 are shown in Tables 5.2, 5.3 and 5.4 respectively. The training results of encoding scheme #5 (BLOMAP) is omitted from Tables 5.3 and 5.4 since the encoded AAs in 5-dimensional vectors for BLOSUM80 and PAM250 were not provided as the encoded AAs were provided in case of BLOSUM62 matrix (Maetschke et al., 2005).

Increasing the dimensions of encoding schemes would increase possibly the training accuracies as shown in Tables 5.2, 5.3 and 5.4. However, the overall result indicates that the combination of the number of dimensions and the type of selected amino acid properties greatly affect the approximation of the substitution matrices such

Table 5.1: The specifications of the fifteen amino acid encoding schemes.

Encoding#	Encoding Scheme Description
1	The presences of nine physicochemical properties of AAs are mapped in a string of 0s and 1s, and converted to an integer (Taylor, 1986).
2	The presences of nine physicochemical properties (Taylor, 1986), and six physical properties called extra tiny, pentagonal, hexagonal, forked and crossed are mapped in a string of 0's and 1's, and converted to an integer (Lac & Kremer, 2009).
3	Based on AA preferences to form the secondary structures (β-strand, α-helix).
4	AAs are represented based on the proposed codon encoding, and each of the four adjacent 0's and 1's is converted to an integer.
5	Based on BLOMAP which converts BLOSUM62 to a 20×5 matrix (Maetschke et al., 2005).
6	AAs are indexed from 1 to 20, and the indexes are represented in 5 bits.
7	The molecular properties of AAs including frequencies, volume, partial specific volume, hydration and *R*-group.
8	Information related to AAs' properties such as exposure to solvent, and the number of C, H, N, O, S atoms.
9	The 7 attributes of Chou-Fasmann's propensity values (Chou et al., 1978).
10	The presences of nine physicochemical properties of AAs are mapped in a string of 0's and 1's (Taylor, 1986).
11	The presences of nine physicochemical properties (Taylor, 1986), and six physical properties called extra tiny, pentagonal, hexagonal, forked and crossed are mapped in a string of 0's and 1's (Lac & Kremer, 2009).
12	The proposed codon encoding scheme represented in a string of sixteen 0's and 1's (Zamani & Kremer, 2011).
13	The combination of all AAs' properties from the three encoding schemes 7, 8, 9.
14	Orthogonal or sparse encoding string (Baldi & Brunak, 2001).
15	The combination of orthogonal (Baldi & Brunak, 2001) and AA polar distribution (Hu et al., 2004).

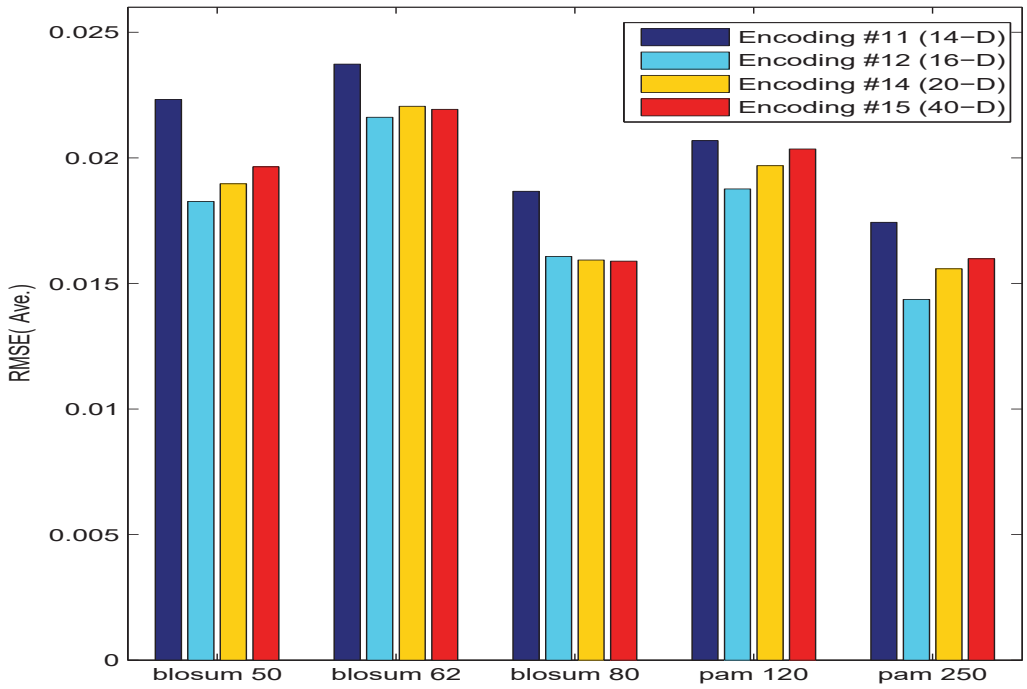

Figure 5.2: The comparison of average *rmse* for encoding schemes #11, #12, #14 and #15 in 10 runs.

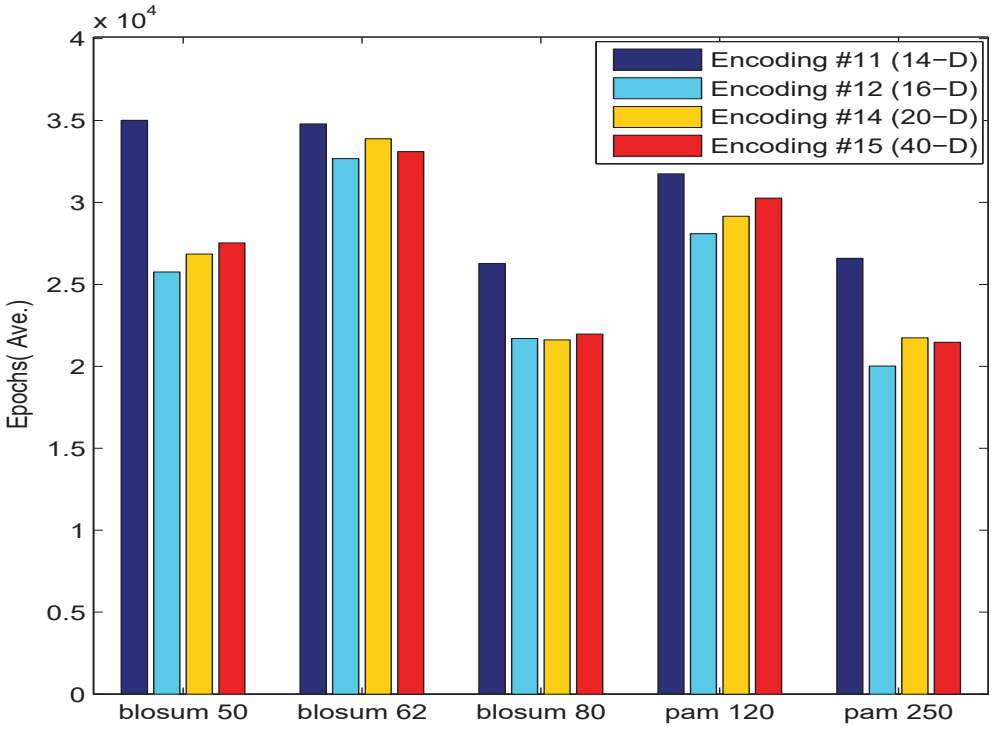

Figure 5.3: The comparison of average training epochs for encoding schemes #11, #12, #14 and #15 in 10 runs.

Table 5.2: The training results of all encoding schemes evaluated by BLOSUM62. The average of root mean square error ($rmse$) for each encoding scheme was calculated in 10 runs.

Encoding#	Dimension	Epoch	Ave. $rmse$
1	1	295	2.504639
2	1	415	2.502990
3	2	175	2.408614
4	4	8910	0.644371
5	5	120	2.286987
6	5	2680	0.580234
7	6	468705	0.350314
8	7	217025	0.900421
9	7	329990	0.253628
10	9	7050	0.693111
11	**14**	**34790**	**0.023730**
12	**16**	**32680**	**0.021617**
13	20	208785	0.282972
14	**20**	**33890**	**0.022051**
15	**40**	**33105**	**0.021930**

Table 5.3: The training results of all encoding schemes evaluated by BLOSUM80. The average of root mean square error ($rmse$) for each encoding scheme was calculated in 10 runs.

Encoding#	Dimension	Epoch	Ave. $rmse$
1	1	265	2.507192
2	1	410	2.503600
3	2	160	2.390761
4	4	7300	0.742286
6	5	1025	0.868625
7	6	492640	0.368557
8	7	341760	0.844570
9	7	371665	0.229421
10	9	5945	0.678370
11	**14**	**26280**	**0.018665**
12	**16**	**21700**	**0.016077**
13	20	349275	0.195062
14	**20**	**21615**	**0.015932**
15	**40**	**21975**	**0.015886**

Table 5.4: The training results of all encoding schemes evaluated by PAM250. The average of root mean square error ($rmse$) for each encoding scheme was calculated in 10 runs.

Encoding#	Dimension	Epoch	Ave. $rmse$
1	1	520	1.904371
2	1	1275	1.896692
3	2	235	1.826631
4	4	18695	0.500904
6	5	2845	0.517455
7	6	417180	0.320489
8	7	612020	0.540101
9	7	267385	0.278978
10	9	5475	0.717099
11	**14**	**26590**	**0.017434**
12	**16**	**20020**	**0.014364**
13	20	338070	0.146217
14	**20**	**21745**	**0.015584**
15	**40**	**21470**	**0.015980**

as encoding schemes #7, #8. Encoding schemes #11, #12, #14 and #15 allow a more precise approximation of the five substitution matrices based on the number of training epochs and *rmse*.

A detailed comparison of the four encoding schemes based on the number of epochs and *rmse* error is shown in Figures 5.2 and 5.3. The experimental result indicate that encoding #12 needs fewer training epochs and leads to a more accurate approximation of the substitution matrices compared to the other three encodings. Also, encoding schemes #12, #14 and #15 perform closely when the substitution matrix is derived from sequences with higher homology such as BLOSUM80. Meanwhile, the performance of encoding #12, improves with substitution matrices derived from sequences with more evolutionary distance such as BLOSUM50 compared to the other three encodings.

The encoding #12 is a better candidate for applications using sequences with low similarity. An overall comparison of the four best encoding schemes based on the average epochs and *rmse* for all five substitution matrices are shown in Figures 5.4 and 5.5. The experimental results also indicate that increasing the dimension of an encoding scheme does not necessarily improve the approximation of the substitution matrices, and meanwhile magnifies "the curse of dimensionality" which is not preferable such as encoding #14 and #15.

The encoding scheme that differentiated better 210 AA pairs and captured the similarities of AA pairs is encoding #12. The better performance of encoding #12 can be explained by knowing that each AA is formed based on a set of genetic codons. Therefore, the different orders and arrangements of the four nucleotides are possibly important to conserve the information related to the underlying similarities and differences among AA pairs. Also, we evaluated the efficiency of a number of amino acid encoding schemes by the training of artificial neural networks to approximate

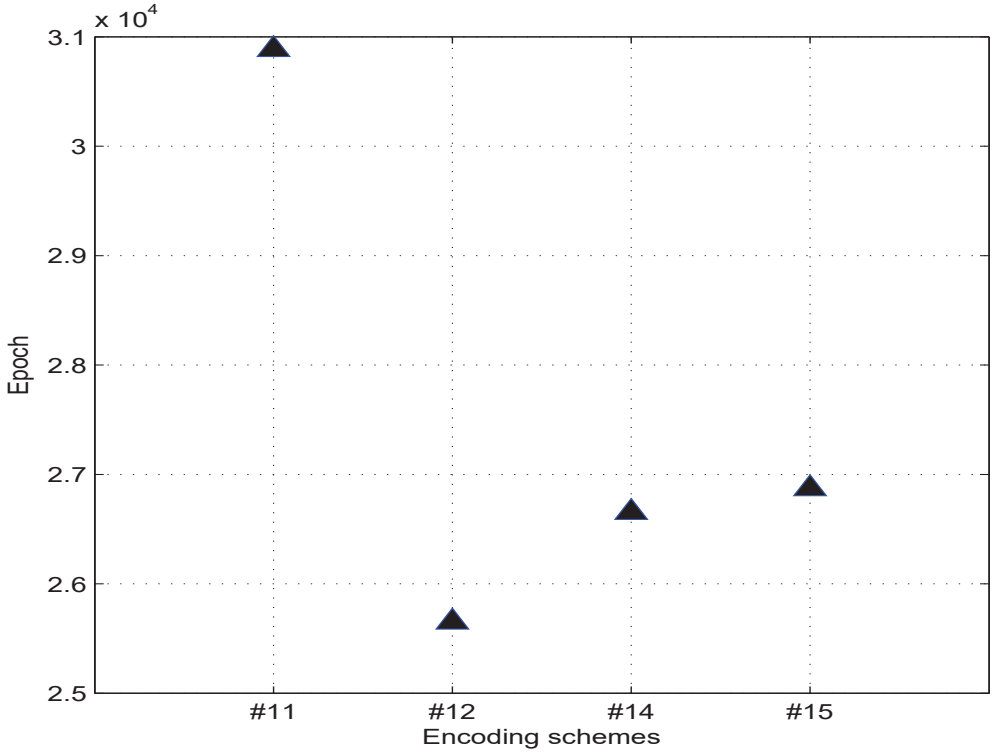

Figure 5.4: The overall comparison of average training epochs for encoding schemes #11, #12, #14 and #15 on all substitution matrices.

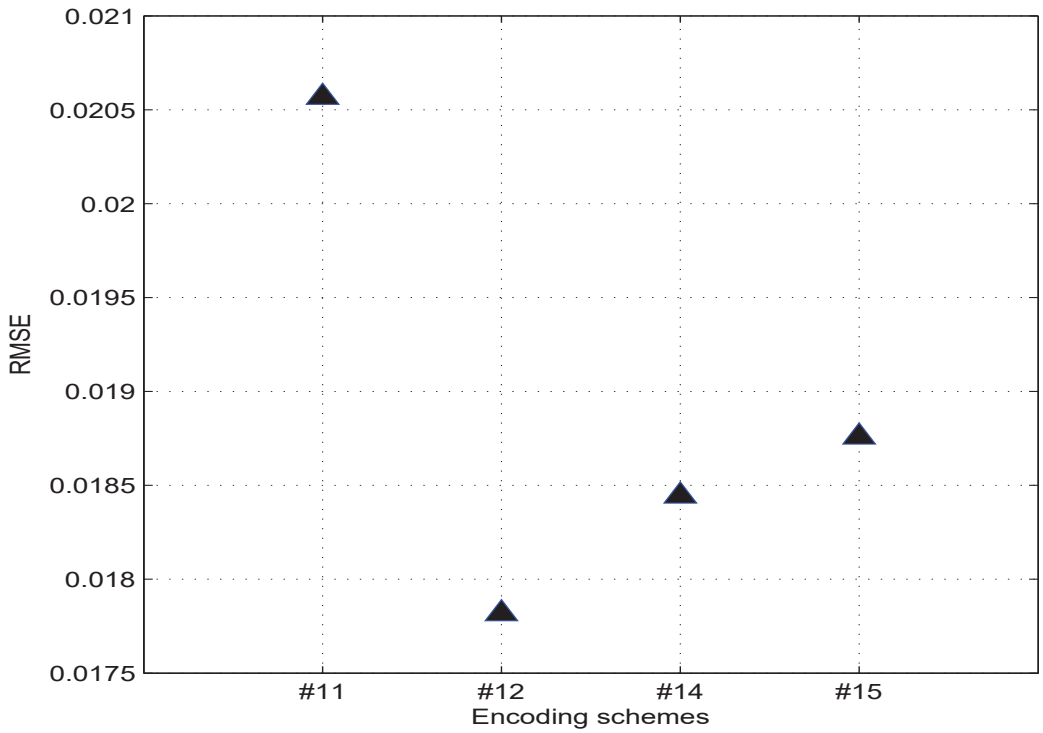

Figure 5.5: The overall comparison of average *rmse* errors for encoding schemes #11, #12, #14 and #15 on all five substitution matrices.

the substitution matrices.

In addition a new amino acid encoding was proposed based on genetic codon mapping concept, and it was compared with a number of commonly used AA encoding schemes. The experimental results indicate the dimension of an encoding scheme and selected AA features are important to the efficiency of the encoding performance. Moreover, we demonstrated an evaluation technique to measure the efficiency of AA encoding schemes which can help us to choose the correct encoding scheme and possibly improve the performance of ML methods in bioinformatics.

The codon encoding scheme is one of the components of the proposed protein secondary structure prediction model which is explained later on in this chapter. The more meticulous evaluation and statistical analyses of the studied encoding schemes, in addition to the approximation of various substitution matrices which is the important step to achieve, are fulfilled by employing the encoding schemes in computational problems related to protein secondary structure prediction and comparing their performances and effects which is discussed in Sections 5.2.4 and 5.5.

5.2.4 PSS Prediction using Codon Encoding

In this section, support vector machines (SVMs) are employed to construct protein secondary structure (PSS) prediction models that incorporate the codon encoding scheme (Zamani & Kremer, 2011) as discussed in Section 5.2.2. The efficiency of codon encoding is compared to that of the commonly used orthogonal encoding scheme for PSS prediction (Zamani & Kremer, 2012). By using orthogonal encoding, a 20-dimensional binary vector is assigned to an amino acid i such that the ith element of the vector is set to 1 and all other elements are set to 0.

In the experiments, SVM classifiers are only trained with encoded AA sequences, and protein profiles are not incorporated in a training to avoid a biased evaluation

of the two encodings. Initially six binary classifiers are constructed based on the two consensus models: (1) one-against-all (OAA) which consists of three binary classifiers as (H/∼H), (E/∼E), (C/∼C), and (2) one-against-one (OAO) which consists of three binary classifiers as (H/E), (E/C), (H/C). For example, classifier (H/∼H) assigns positive class labels to the residues in helices and negative class labels to the residues in strands and coils, and classifier (H/E) assigns positive class labels to residues in helices and negative class labels to the residues in strands.

Moreover, five tertiary classifiers are constructed with different combinations of the six binary classifiers as follows (Hua & Sun, 2001):

1. SVM_MAX_D tertiary classifier which consists of all three binary classifiers based on the OAA scheme. An AA segment is tested by the binary classifiers, and the classifier with the largest positive distance to optimal separating hyperplane (OSH) determines the predicted secondary structure of the central residue in the AA segment.

2. SVM_TREE1 tertiary classifier which consists of two binary classifiers, one from the OAA scheme and the other from the OAO scheme. For instance, a sampling residue is tested by H/∼H and if the OSH value is positive, the residue is assigned to a helix. Otherwise the residue is tested by the classifier (E/C) and, similarly, if the OSH value is positive the sampling residue is eventually assigned to a strand, otherwise to a coil. Similarly, the tertiary classifiers SVM_TREE2, SVM_TREE3 are constructed using the classifier (E/∼E, H/C) and (C/∼C, H/E) respectively.

3. SVM_VOTE tertiary classifier which consists of all six binary classifiers and the secondary structure with the highest number of counts is considered as the final prediction.

4. DAG tertiary classifier which is a directed-acyclic graph and cascades the three binary classifiers from the OAO scheme (Nguyen et al., 2003). For example, if an OAO classifier that predicts secondary structure E against secondary structure H and the prediction result is not secondary structure E, an OAO classifier which predicts secondary structure H against secondary structure C is used in the second tier.

5. SVM_JURY which counts the number of predicted secondary structures of all tertiary classifiers from 1 to 4, and the final predicted secondary structure is the one with the maximum number of votes.

The commonly used protein datasets to evaluate PSS prediction methods are RS126 (Rost et al., 1993) and CB513 (Cuff & Barton, 1999) which consist of 126 and 513 nonhomologous protein chains respectively. The sequence identity measurement in the datasets is based on the stringent definition (SD) of sequence similarity by which sequence alignments whose SD scores are greater than or equal to 5 are considered to be identical. Moreover, the similarity of each sequence pair is less than 25% when the lengths of the paired sequences reach over 80 residues. A throughout evaluation of PSS prediction methods relies on the degree of pairwise similarity in datasets. Otherwise, high pairwise similarity implies an overfitting and a weak generalization of a learning model. We pointed out the limitations related to the homology issue in a number of studies in Chapter 2.

The dataset RS126 contains approximately twenty four thousand residues with an average of 185 residues per sequence. The dataset CB513 has 513 non-homologous sequences which contains over one hundred and twenty thousand residues with an average of 180 residues per sequence. The tertiary structures of the protein chains in the datasets were obtained from Protein Data Bank (PDB) (Sussman et al., 1998). The secondary structures of the protein sequences were determined by the DSSP

method which is a the most commonly used and standard method for protein secondary structure assignments (Kabsch & Sander, 1983) as explained in Chapter 3.

In the following experiments, we used the three-state reduction scheme applied in PHD (Rost, 1996) as follows: (1) I, G and H to H, (2) B and E to E, (3) the rest to C. Initially six binary SVM classifiers as described earlier in this section were trained using LibSVM (Chang & Lin, 2011). The binary SVM classifiers incorporate the RBF kernel function which is efficient for most nonseparable input spaces as follows:

$$K(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2) \quad , \quad \gamma > 0 \tag{5.2}$$

where x_i, x_j are n-dimensional vectors in the input space, and γ is a predefined parameter. In SVM-based methods for PSS prediction, it has been shown RBF kernels are robust in generalization and convergence speed (Hua & Sun, 2001; Kim & Park, 2003). The kernel parameter γ was set to $\frac{1}{k}$ where k is equal to the number of features in input space, and with a sliding window of length l and an n-dimensional encoding, k is equal to $l \times n$. The generalization parameter C was set experimentally to 1.5.

In order to make sure that the experimental results of classifiers using the codon encoding scheme are comparable with those of other PSS prediction methods (Hua & Sun, 2001; Nguyen et al., 2003), we applied a 7-fold cross validation on the RS126 dataset, and the window lengths were chosen with 11, 13 and 15 residues. Thus, each binary SVM classifier is trained with six folds and the remaining fold is used for a validation test. AA segments defined by scanning a sliding window of a fixed length on the protein sequences are encoded using codon and orthogonal encodings. If the start and end sections of AA sequences are scanned, a sliding window may not contain an exact predefined number of residues. Therefore, an extra dimension is added to both encodings to discriminate the $N-$ and $C-$ terminus. As a result, each encoded residue is in the form of 17- and 21-dimensional binary vectors using codon

and orthogonal encodings respectively.

Next, we constructed eight tertiary classifiers (1-6 (Hua & Sun, 2001), 7-8 (Nguyen et al., 2003)) by various combinations of the trained binary classifiers as outlined in Table 5.6. Tertiary classifier #2 was built by combining all six binary classifiers whereas tertiary classifier #8 was built by combining only the three binary classifiers based on the OAO scheme. The overall accuracy of the tertiary classifiers are measured using Q_3 and *SOV99* (Zemla et al., 1999) where the latter measures the segments of correctly predicted secondary structures and the former accounts for correct prediction of a residue's secondary structure. In the experiments, the binary and tertiary SVM classifiers are trained only with AA sequences since we aimed to evaluate the effectiveness of the classifiers without using additional data such as protein global and evolutionary information. The experimental results are organized in two groups: (1) the binary SVM classifiers, and (2) the tertiary SVM classifiers. The classification performance of the six binary classifiers using various lengths of a sliding window is shown in Table 5.5.

Table 5.5: The classification results of binary classifiers using codon (cod.) and orthogonal (orth.) encoding schemes and different window lengths (L) on RS126 dataset (Zamani & Kremer, 2012). For each classifier, the highest classification accuracy obtained with a specific sliding window's length is shown in boldface.

Binary classifier	L=11		L=13		L=15	
	Cod.	Orth.	Cod.	Orth.	Cod.	Orth.
C/∼C	71.70	70.80	**71.85**	**71.00**	71.85	70.95
E/∼E	79.80	78.75	79.92	78.72	**80.25**	**78.85**
H/∼H	76.55	73.90	76.75	74.25	**78.15**	**74.30**
C/E	**60.20**	**60.15**	60.00	60.05	59.90	60.05
E/H	**69.30**	**68.45**	69.05	67.80	69.00	68.00
H/C	71.85	69.90	72.35	70.15	**72.60**	**70.30**

The results indicate the performance of binary classifiers varies with different

lengths of sliding window. The binary classifiers built based on the OAA consensus scheme performed better than the binary classifiers built based on the OAO scheme as shown in Figures 5.6 and 5.7.

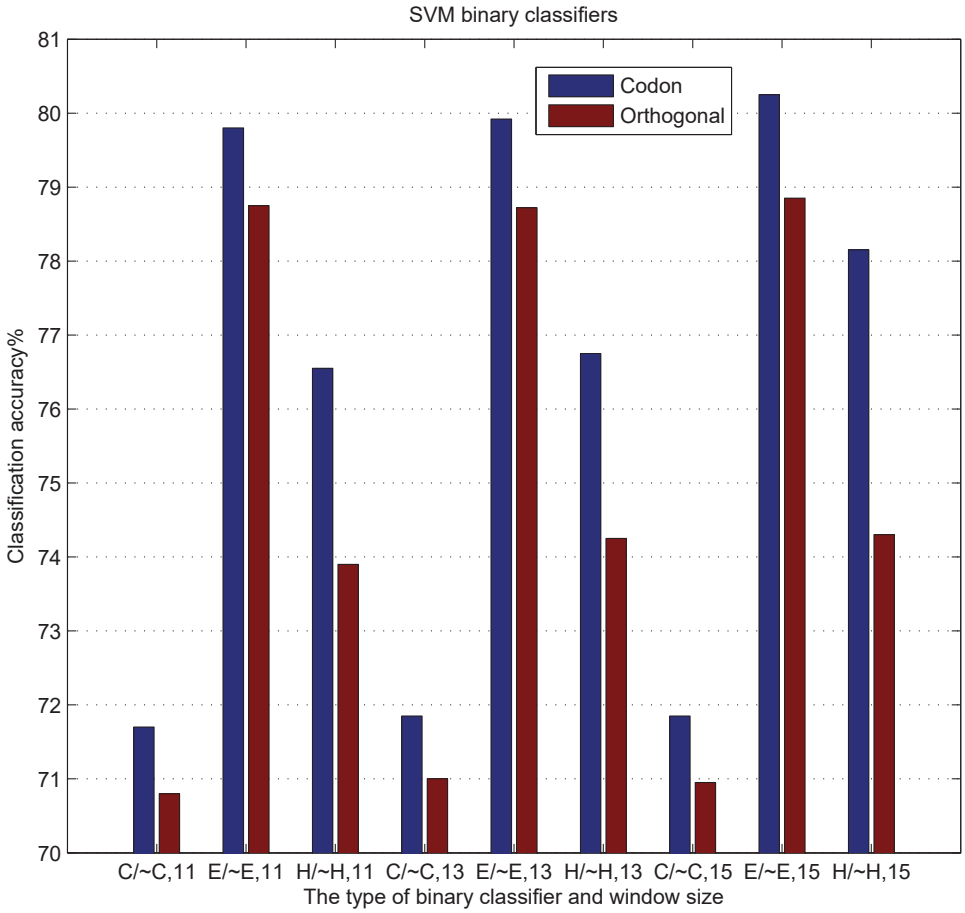

Figure 5.6: The classification accuracy on binary SVM classifiers using codon and orthogonal encodings based on one-against-all scheme (Zamani & Kremer, 2012).

The superior performance of the OAA classifiers is possibly due to the higher number of training instances compared to those of the OAO classifiers. By using the OAA scheme, the classifier C/∼C results in the lowest classification accuracy whereas using

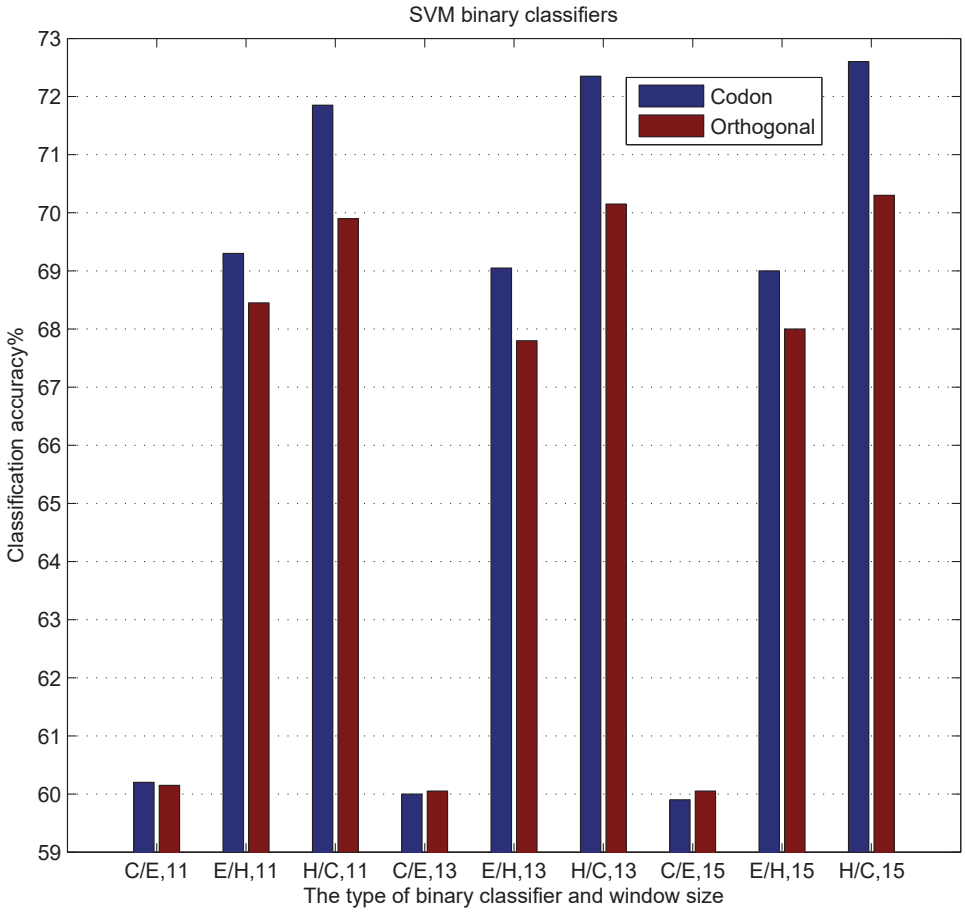

Figure 5.7: The classification accuracy on binary SVM classifiers using codon and orthogonal encodings based on one-against-one scheme (Zamani & Kremer, 2012).

the OAO scheme the classifier C/E has the lowest performance. In comparison to the orthogonal encoding, applying the codon encoding improved the prediction performance of all binary classifiers except for the classifier C/E where the performance of the codon and orthogonal encodings are relatively identical if the sliding window contains 13 and 15 residues. Thus, it can be concluded that the differentiation of strand and coil classes is a difficult classification task since the prediction accuracy of the classifier C/E was lower than those of the other classifiers which incorporate the same encoding schemes in the experiment. In addition, the maximum classification performances of the two types of the binary classifier are achieved when the window sizes are identical regardless of the selected encoding schemes as shown in Table 5.5.

The overall performance of the eight tertiary classifiers using the codon and orthogonal encodings is shown in Table 5.6. The tertiary classifiers were constructed by the binary classifiers that achieved the highest accuracy by specific window sizes as highlighted in Table 5.5. Using the codon encoding increases the Q_3 scores by up to 2.5% and the SOV scores by up to 3.5% compared to the orthogonal encoding.

The performances of the tertiary classifiers using the codon encoding are compared to those of structurally similar tertiary classifiers, except the classifiers 4 and 5, using orthogonal encoding and protein profiles as shown in Table 5.7. The tertiary classifiers 1-5 (Nguyen et al., 2003) and 9-11 (Hua & Sun, 2001) incorporated protein profiles in the trainings. The tertiary classifiers 4, 5 are multi-class SVM classifiers which performed better than the cascaded binary SVM classifiers 9-11 (Weston & Watkins, 1999; Crammer & Singer, 2002). The performance of the tertiary classifiers 6-8 using only the codon encoding are comparable with the other classifiers that incorporated additionally profile information as shown in Table 5.7. The Q_3 scores of the classifiers 6-8 are competitive with those of the classifiers 9-11.

Similarly, the SOV scores of the classifiers 6-8 surpassed the SOV scores of the

Table 5.6: The classification performance of tertiary classifiers using the codon and orthogonal encoding schemes on RS126 dataset (Zamani & Kremer, 2012). The classifier #2 was built by combining the six binary classifiers of OAO and OAA schemes whereas the classifier #8 was built by combining the three binary classifiers of the OAO scheme.

#	Tertiary classifier	Codon		Orthogonal	
		Q_3 (%)	SOV (%)	Q_3 (%)	SOV (%)
1	SVM_MAX_D	67.5	60.3	65.7	57.5
2	SVM_VOTE[1]	66.1	61.7	65.0	57.9
3	SVM_TREE1	65.0	60.5	64.0	59.2
4	SVM_TREE2	66.2	54.0	64.8	53.1
5	SVM_TREE3	65.7	59.7	63.4	56.1
6	SVM_JURY	66.1	61.7	65.0	57.9
7	DAG	67.1	62.3	64.9	59.5
8	SVM_VOTE[2]	66.8	61.7	64.9	59.5

Table 5.7: Overall comparison of various tertiary classifiers using protein profiles and AA encoding on RS126 dataset (Zamani & Kremer, 2012). The classifiers #2 and #7 incorporated orthogonal and codon encodings respectively and were built by combining the binary classifiers of the OAO scheme.

#	Tertiary classifier	Q_3(%)	SOV(%)	Profile	AA encoding
1	SVM_MAX_D	70.4	59.0	✓	Orthogonal
2	SVM_VOTE[1]	70.1	58.3	✓	Orthogonal
3	DAG	70.1	58.3	✓	Orthogonal
4	VW	70.5	57.1	✓	Orthogonal
5	CS	70.4	56.5	✓	Orthogonal
6	SVM_MAX_D	**67.5**	**60.3**	–	Codon
7	SVM_VOTE[2]	**66.8**	**61.7**	–	Codon
8	DAG	**67.1**	**62.3**	–	Codon
9	SVM_TREE1	68.0	69.5	✓	Orthogonal
10	SVM_TREE2	67.5	69.1	✓	Orthogonal
11	SVM_TREE3	66.8	68.6	✓	Orthogonal

classifiers 1-5. Lastly, the dimension of the codon encoding scheme is less than that of the orthogonal encoding scheme. Using an encoding with fewer dimensions is an advantage in high-dimensional input spaces and reduces training time. In the experiments, we aimed to solely evaluate the efficiency of the codon encoding scheme according to local information. The training of SVM classifiers with the codon encoding resulted a competitive classification performance in comparison to the similar SVM classifiers' using protein profiles and the orthogonal encoding.

The codon encoding implicitly incorporates partial protein evolutionary information since the encoding is based on the genetic codon mappings at the DNA level prior to the AA formations which has a domain with less dimensional complexity compared to that of the twenty AAs. In Section 5.3, the efficiency of the codon encoding scheme in protein secondary structure prediction is examined further using larger protein datasets such as CB513 (Cuff & Barton, 1999).

5.3 GP-based PSS Model using Clustering

In this section, we propose an evolutionary computation model for protein secondary structure (PSS) prediction which applies a new encoding technique for encoding amino acid (AA) sequences by using clustering (Zamani & Kremer, 2015b). The performance of the proposed GP-based model is compared to those of feed-forward artificial neural networks (ANNs) and support vector machines (SVMs) which incorporate separately AA sequences and clustering information. In the aforementioned methods, we adopted a 1-tier architecture for the three multi-class prediction models, and additional information such as proteins' profiles and structures is not incorporated with the classifiers in the learning and evaluation processes to measure solely the performance of the applied GP model and the effect of the information derived

from clustering. We use the information of proteins' profiles in the experiments which are performed in the next sections. Moreover, the structural information of proteins is not incorporated directly in the experiments since our proposed secondary structure prediction model is based on an *ab initio* prediction scheme.

Genetic programming (GP) (Koza, 1992) is an evolutionary computing technique which has been successfully applied in a number of optimization problems. The great advantages of evolutionary methods are the exploitation and exploration of a search space which are crucial for exploring a high dimensional search space with many local minima. In the GP model, a genotype (tree) is represented as one or multi-nested "IF" rules as shown in Figure 4.11, and the sets of logical and arithmetic functions, and terminals are shown in Table 4.1.

The logical functions return *true* or *false*, and the arithmetic functions return numeric values. The "B" function has three arguments a,b,r, and if r is within $[a,b]$, the function returns *true*, otherwise *false*. The "C" terminal has two parameters c, w which are a class label and the weight of the class label respectively. The "S" terminal has two indices l_1, l_2 which are the indices of locations on an AA segment with length L.

For a given sequence segment, a tree evaluation starts from the leaf nodes "S", "C". An individual tree is traversed and evaluated by using a bottom-up technique as is explained in detail later on. An "IF" function has three arguments which are a logical function and two "C" terminals, and updates a certainty vector denoted *cv*. Certainty vector *cv* has three elements whose values are updated during a tree evaluation. In an "IF" function, if the conditional argument is *true*, the ith element of *cv* is updated, otherwise the jth element. The indices i, j are class labels of the "C" terminals whose weight values are used to update *cv*. The first argument (conditional branch) of an "IF" function is evaluated prior to "C" terminals. If an "IF" function

is a child node, the label of a winning class is passed to the parent.

The node types of the trees should be preserved by the genetic operators when genetic operators are applied on a population, otherwise the outcome would be a tree with incorrect grammar. An "IF" function can be changed to a "C" terminal and vice versa. A mutation operator can change the indices l_1, l_2 of an "S" terminal to numeric values within [1,L]. However, the l_2 index, 50% of the time is changed to a number within $[1, 20]$ where the upper bound number corresponds to the twenty amino acids. Thus, there are two scenarios for evaluating an "S" terminal: (1) if indices l_1, l_2 are within $[1, L]$. The residues located at the positions l_1, l_2 of an AA segment are identified and their evolutionary distance is defined by using a substitution matrix (Henikoff & Henikoff, 1992), (2) if index l_2 is within $[1, 20]$, the evolutionary distance of the AA with the index l_2 and residue located at the position l_1 of the sequence segment is defined using a substitution matrix. Moreover, similar to the mutation, the recombination of two trees must preserve the node types of the branches which are swapped to avoid generating trees with incorrect syntax.

In the experiments, the performances of PSS classifiers are measured by using the two standard protein datasets which are RS126 (Rost et al., 1993) and CB513 (Cuff & Barton, 1999). The description of the datasets that contain protein sequences with pairwise similarity of below 25% were explained previously in detail in Section 5.2.4. As was mentioned earlier, a learning model that uses high similar sequences can be limited to a weak generalization and an overfitting issue. We investigate the performance of a PSS prediction model trained by using sequences with higher pairwise identity in Section 5.5. Similarly, the 3D structures of the protein sequences were extracted from Protein Data Bank (PDB) (Sussman et al., 1998), and the secondary structures of the protein sequences were defined by the DSSP program (Kabsch & Sander, 1983).

Moreover, there are various methods for the eight- to three-state reduction. In the experiments, the three-state reduction scheme is based on the PHD secondary structure assignment (Rost, 1996). The experiments were performed in two parts: (1) the ANN, SVM and GP classifiers are evaluated by using AA sequences, (2) the performance of the three classifiers are examined by using the information extracted from a clustering technique. Initially, AA sequences are segmented by moving a sliding window of a fixed length along the sequences. The AA segments are encoded by using the codon encoding described in Section 5.2. A 5-fold cross validation technique is used throughout the experiments. We set a window length of 11 residues which is chosen based on the experimental results in previous sections. Thus, the ANN and SVM classifiers have 11×17 input units. An additional dimension was added to the codon encoding which corresponds to the end and beginning of a sequence.

The proposed GP-based classifier incorporates the entire section of an AA segment, however, the residues whose positions defined by "S" terminals participate in an evaluation process as explained earlier in this section. In the experiments, a feed-forward, multilayer ANN is employed which has two hidden layers and three output units in the output layer corresponding to three secondary structures H, E and C. The SVM classifier is a multiclass SVM (Tsochantaridis, Joachims, Hofmann, & Altun, 2005). In the GP classifier, tournament selection (Goldberg & Deb, 1991) is used. In the tournament selection, n individuals are randomly chosen from a population, and n is the tournament size.

In the experiments, two individuals are chosen each time to reduce the computational complexity. The individuals in a selection pool compete and the winner, fittest individual, is passed to the next generation. The population size which is set to 64 was chosen based on the available computational resources. In each generation, the mutation operator is applied to all selected individuals, and 85% of the time the

fittest individual is selected after the crossover operator is applied on the individuals in the selection pool. In the remaining 15% of the time, the mutated individuals are passed directly to the next generation. In this way, the search strategy allows both the exploitation and exploration of the solution space and keeps the track of the genetic diversity in the population.

In the second part of the experiments, after AA segments are encoded to a set of numeric vectors, we use the k-means clustering algorithm to cluster the segments. The clustering technique is applied on the segments that are selected for a training set. Ten cluster centers were chosen experimentally as the predefined number of clusters. In each cluster, the probabilities of the three secondary structures are calculated based on the central residues of the AA segments within the cluster. For an AA segment, the distance between the segment and the centers of the clusters are calculated and the segment is assigned to the closest cluster.

After identifying the clusters of each segment, the central residue of the segment is represented to the classifiers by three probabilities. Therefore, the number of input units in the ANN and SVM classifiers is equal to $3 \times l$ where l is the length of an sliding window. In the GP classifier, as explained earlier, the output of an "S" terminal is computed based on the evolutionary distance of two identified residues.

However, when an AA segments is represented by the probabilities derived from clusters, 3-dimensional vectors are assigned to the residues located at the positions l_1, l_2, and then the Euclidean distance of the vectors is computed as illustrated in Section 4.2. However, if AA segment s is in cluster c and index l_2 does not define a position on segment s, then the probability values related to amino acid k at position l_1 in cluster c are assigned to index l_2 where k is equal to l_2. The performances of the three PSS prediction methods were initially evaluated by using AA sequences and the experimental results are shown in Tables 5.8 and 5.9.

Table 5.8: The performance of PSS classifiers evaluated by using amino acid sequences from dataset RS126 (Zamani & Kremer, 2015b).

Method	Q3(%)		SOV99(%)
	Ave.	Max.	
ANN	61.7	63.5	60.4
SVM	62.5	64.6	61.3
GP	63.7	65.8	62.1

Table 5.9: The performance of PSS classifiers evaluated by using amino acid sequences from dataset CB513 (Zamani & Kremer, 2015b).

Method	Q3(%)		SOV99(%)
	Ave.	Max.	
ANN	62.2	64.1	63.2
SVM	63.4	65.3	62.9
GP	65.0	66.9	64.5

The performances of the PSS prediction methods that are evaluated by using clustering information are shown in Tables 5.10 and 5.11. The second column in the tables include two $Q3$ scores: (1) the average $Q3$ score is calculated from the prediction results of each classifier by using a 5-fold cross-validation technique, and (2) the maximum $Q3$ score is the highest prediction accuracy that each classifier achieves in five folds. The training of the GP classifier is performed in 10 runs, and the trained GP model that achieves the highest $Q3$ score is selected for a validation test. The last column shows the SOV score which is computed when the maximum $Q3$ score is achieved in a run. As was previously discussed, incorporating AA properties and proteins' global and evolutionary information improves the prediction performance. However, the additional information is not included in order to measure the effect of clustering information in the experiments.

The performance of the classifiers are higher when dataset CB513 is used based on

Table 5.10: The performance of PSS classifiers evaluated by using clustering information derived from dataset RS126.

Method	Q3(%)		SOV99(%)
	Ave.	Max.	
ANN	63.9	65.7	63.1
SVM	64.6	66.2	63.5
GP	65.1	67.3	64.8

Table 5.11: The performance of PSS classifiers evaluated by using clustering information derived from dataset CB513.

Method	Q3(%)		SOV99(%)
	Ave.	Max.	
ANN	64.2	66.3	64.5
SVM	65.1	67.8	65.0
GP	66.4	69.0	67.1

the prediction scores. The results indicate the PSS classifiers capture the patterns of various AA combinations, which contribute the formation of secondary structures, if the classifiers are trained with a higher number of sequences. On average, the prediction scores of the GP classifier is 3% higher than those of the ANN and SVM models. In addition, the prediction accuracy of the classifiers that incorporate clustering information are 2% higher than those of the similar classifiers which incorporated AA sequences.

The difference between the average and maximum Q_3 scores is based on two causes: (1) various internal parameters of the classifiers in different runs, e.g. weights in ANNs, randomly generated genotypes in the GP method, (2) the random selection of training instances to balance the number of class labels to prevent overfitting of a specific class label in a training. Although, the Q_3 and SOV scores are interpreted differently and not related, the maximum Q_3 scores are higher than SOV scores

as usual. However, the average Q_3 scores are close to SOV scores based on the experimental results.

The experimental results indicate the proposed PSS prediction model based on GP outperformed the feed-forward multilayer ANN and multi-class SVM classifiers. Also, the classifiers that incorporated statistical information from clustered AA segments performed on average 3% higher than the similar classifiers using AA sequences. In addition, the number of input units in a classifier is reduced if the information derived from clusters are represented to the classifiers.

5.4 Two-stage PSS Model and Information Theory

In this section, we propose a two-stage PSS prediction model based on a hybrid Machine Learning technique (Zamani & Kremer, 2015a). The two-stage prediction model incorporates protein profiles and statistical information which are extracted from protein sequences by using a k-means clustering and Information theory. The performance of the proposed PSS prediction method is compared to those of two-tier feed-forward ANNs and SVMs which incorporate protein profiles. As was discussed in Chapter 2, PSS prediction performance was improved by using cascaded architectures, consensus techniques, and protein evolutionary information (Hu et al., 2004). In the consensus technique, a voting scheme is applied to the outcomes of multiple PSS prediction methods which may have different or similar designs.

In this study, the aim is to examine individually the performances of the two-stage PSS model and two cascaded PSS classifiers based on ANNs and SVMs which are used as a benchmark. The two-stage PSS prediction model employs multilayer feed-forward ANNs and a genetic programming (GP) method. The overall scheme of

the two-stage prediction model is shown in Figure 5.8.

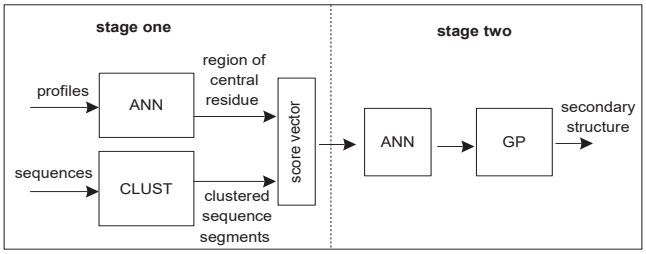

Figure 5.8: An overall scheme of the two-stage PSS prediction model. CLUST represents the component of k-means clustering.

In the first stage, the profiles of an amino acid (AA) segment are mapped to a region on the Ramachandran plot (2D-plot) (Ramachandran & Sasisekharan, 1968) by a feed-forward ANN. An AA segment consists of a predefined number of residues, and it is defined by moving a sliding window along sequences. A *score* vector is constructed for the central residue of an AA segment based on the identified region on the 2D-plot and statistical information derived from clusters and Information theory. A score vector represents the preferences of the two middle residues of an AA segment for adapting all two possible combinations of the three-state secondary structures in a region on the 2D-plot. The construction of a score vector is explained in detail in Section 4.2.

In the second stage, a feed-forward ANN maps the score vectors of an AA segment to all intermediate (likelihood) three-state secondary structures. Next, the GP method determines the final secondary structure prediction based on the intermediate three-state secondary structures of the AA segment. The design, training and evaluation of the proposed GP model are explained in detail in Sections 3.4 and 5.3.

Moreover, we used the dataset RS126 (Rost et al., 1993) and a 5-fold cross validation technique was used throughout the experiments in this section. AA sequences

were encoded by using the Codon encoding method (Zamani & Kremer, 2011) before applying a k-means clustering on the dataset. The description of the dataset, the encoding scheme and clustering are provided in details in Sections 5.2 and 5.3.

In the first stage, the dihedral angles of protein backbones were calculated from atomic coordinates defined in the PDB (Sussman et al., 1998). The protein profiles were extracted from the HSSP dataset (Sander & Schneider, 1991). Initially, we determined the eight classes of protein secondary structures by using the DSSP program (Kabsch & Sander, 1983) and reduced them to three classes based on PHD scheme (Rost, 1996) as explained in Section 3.6. The Ramachandran plot is divided in four regions which means dihedral angles ϕ, ψ are grouped in the ranges of $[-180, 0]$, $[0, 180]$. Then, a feed-forward ANN was employed to map the profiles of AA segments to one of the four defined regions in the 2D-plot. The ANN has two output units and $20 \times l$ inputs where l is the length of an AA segment which was set to eleven residues.

In the clustering component denoted CLUST, AA sequences are segmented and encoded, and then they are clustered. For an AA segment, the distance between the segment and the centers of the clusters are calculated and the segment is assigned to the closest cluster. Ten clusters were used in the experiments, and the number of clusters was chosen experimentally based on initial outcomes. Initially, for a target segment that is assigned to a cluster, a set of AA segments similar to the target segment is identified in the cluster by using a nearest-neighbour technique (Salamov et al., 1995). Then, score vectors are constructed for the target segment according to Equations (4.1) and (4.2).

In the second stage, a feed-forward ANN is employed, and it incorporates the score vectors of AA segments. The neural network has $9 \times l$ input units where l is equal to the length of an AA segment. In the GP model, a two-tournament selection (Goldberg & Deb, 1991) is used such that the winner (fittest individual) is selected for the next

generation. The population size which is equal to 64 was chosen based on the available computational resources. In each generation, the mutation operation is applied on all selected individuals, and the crossover rate was chosen 85% in the selection pool as explained in Section 5.3.

In the GP classifier, as explained in Section 5.3, the return value of an "S" terminal is computed based on the Euclidean distance of two 3-dimensional vectors that are assigned to the residues located at the positions l_1, l_2 of an AA segment. The experimental results of the proposed multi-stage classifier are compared to those of two-tier PSS classifiers which commonly employ ANNs and SVMs. The PSS prediction methods that were selected in this study are based on the cascaded ML architectures, similarity on the applied generalization technique, and the training and test datasets. The performance of the three classification models are represented in Table 5.12. The columns C_H, C_E, C_C represent the correlation coefficient values of the prediction results for the three secondary structures H, E, C.

Table 5.12: The comparisons of the three PSS classifiers using RS126 dataset. MGP denoted for the multi-stage GP classifier.

Method	Q3(%)	C_H	C_E	C_C	SOV(%)
ANN (Rost, 1996)	70.8	0.60	0.52	0.51	73.5
SVM (Hua & Sun, 2001)	71.2	0.61	0.51	0.52	74.6
MGP	74.5	0.67	0.54	0.56	76.9

The experimental results indicate the proposed multi-stage classifier improves 3% the Q_3 score and 2% the SOV score in comparison to those of two-tier classifiers that employ ANNs and SVMs. The Q_3 score obtained from the GP classifier is the maximum accuracy observed in 10 runs. In addition, the correlation coefficient values indicate helices are predicted more accurately than coils and strands. In this study, the goal was to represent the prediction results that are obtained from stand-alone

methods which use the cross-validation technique. Therefore, a higher prediction accuracy could be achieved by using PSS classifiers that employ consensus techniques such as web-based PSS prediction servers (Mirabello & Pollastri, 2013).

Moreover, using two different datasets for a training and a test, or dataset that are compiled with less restrictive similarity rules were excluded in the study because these choices could result in classifiers which are biased toward the PSS prediction of closely related sequences rather than those of distant sequences. As was mentioned previously, a learning model that incorporates highly similar sequences can be limited to a weak generalization and an overfitting problem. The performance of a PSS prediction model incorporated with sequences that have higher pairwise identity than twilight zone (25%) is examined in Section 5.5.

As shown in Table 5.12, the experimental results indicate our proposed hybrid PSS prediction model outperforms cascaded ANN and SVM classifiers. Also, in the proposed method, we used the properties of Ramachadran plot and incorporated statistical information, which were derived by using clustering and Information theory, in the classification process. In addition, the number of required input units of a classifier in the second stage is reduced by using the score vector scheme.

5.5 Two-stage PSS Model and Transition Sites

In this study, we developed the two-stage PSS prediction model proposed in Section 5.4. The extended two-stage ML model predicts secondary structures through a novel framework of PSS transition sites (Zamani & Kremer, 2016). PSS transition sites are locations on AA sequences when one type of secondary structure is changed to another type of secondary structure as shown in Figure 4.2. The the overall scheme of the two-stage PSS prediction model is shown in Figure 4.1.

In this section, we mainly discuss the experiments that were performed by using the two-stage PSS prediction model which consists of a novel framework of PSS transition sites and the PSS secondary structure prediction model as shown in Figures 4.3 and 4.8. The proposed model is illustrated comprehensively in Section 4.2. The performance of the proposed method is compared to the state-of-the-art cascaded ML methods which commonly employ ANNs and SVMs. The evaluation is performed in two steps: (1) PSS transition site prediction, (2) PSS structure prediction.

A common generalization technique to evaluate the performances of PSS classifiers is based on an n-fold cross-validation technique where n is the number of partitions. By this approach, protein datasets that contain a few hundred protein sequences are chosen with regards to computational costs. As a result, the overall prediction accuracy calculated in this way is affected by two restrictions: (1) a small number of partitions, and (2) datasets with relatively small sizes.

In addition, a comprehensive evaluation of a PSS prediction method relies on the degree of similarity between the training and test sequences which means the identity of every two sequences is required to be less than the "twilight zone" which is 25%. Otherwise, it results in an overfitting and a weak generalization of the method's performance in case of distant sequences. We pointed out the aforementioned limitations in a number of studies in Chapter 2.

In this study, we selected a very large set of the latest nonhomologous protein sequences, PISCES (Wang & Dunbrack, 2003) which contains approximately nine thousand chains and two million residues, and the identity of every sequence pair is less than 25%. In addition, the statistical significance of the experimental results are measured by a Wilcoxon rank-sum nonparametric test (Wilcoxon, Katti, & Wilcox, 1970) since the distribution of Q_3 scores from the experimental results when is examined by using the Shapiro-Wilk normality test (Shapiro & Wilk, 1965) does not follow

a normal distribution. Evaluating a PSS prediction method by using the statistical test and analyzing the distribution of Q_3 scores provide a more comprehensive way of the method's performance, and we suggest the statistical evaluation technique to be adopted in other studies related to PSS prediction techniques.

Moreover, each statistical test is performed on five hundred test samples. A test sample contains five thousand residues which are selected randomly from the portion of the dataset that is assigned for a validation test. Initially, the prediction performance of the proposed two-stage classifier which did not incorporate PSS transition site was compared to those of two-tier ANN and SVM classifiers. The comparison of the prediction performance based on Q_3 scores is shown in Figure 5.9.

In order to examine the performance effectiveness of PSS transition sites on the proposed classifier, the prediction performance of the classifier without transition sites is compared to that of the classifier with transition sites. The distribution of Q_3 scores from the prediction results of the two classifiers are shown in Figure 5.10.

The experimental results and statistical analyses indicate that the distribution of Q_3 scores by the proposed two-stage prediction model without using PSS transition sites is approximately 3.5% and 2.5% higher than those of the two-tier ANN and SVM classifiers respectively, and the p-values of the statistical tests are less than 0.001 by the Wilcoxon rank-sum test as shown in Figure 5.9.

The distribution of Q_3 scores is increased in overall 6% with a p-value that is less than 0.001 by the Wilcoxon rank-sum test when the proposed two-stage classifier incorporates the information of PSS transition sites as shown in Figures 5.9 and 5.10. In addition, incorporating the information of transition sites reduces the number of Q_3 scores that are considered outliers in the distribution. The distribution of Q_3 scores of the cascaded SVM classifier consists of fewer outlier scores than that of the two-tier ANN classifier as shown in Figure 5.9.

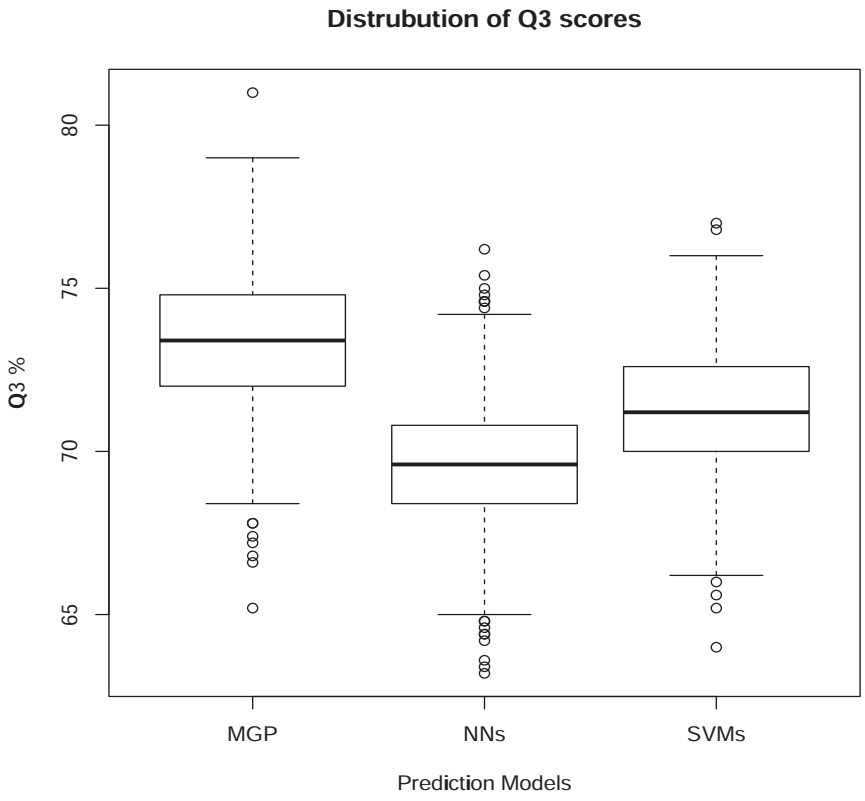

Figure 5.9: The performance of the proposed multi-stage PSS prediction model (MGP) and 2-tier ANNs and SVMs on dataset PISCES.

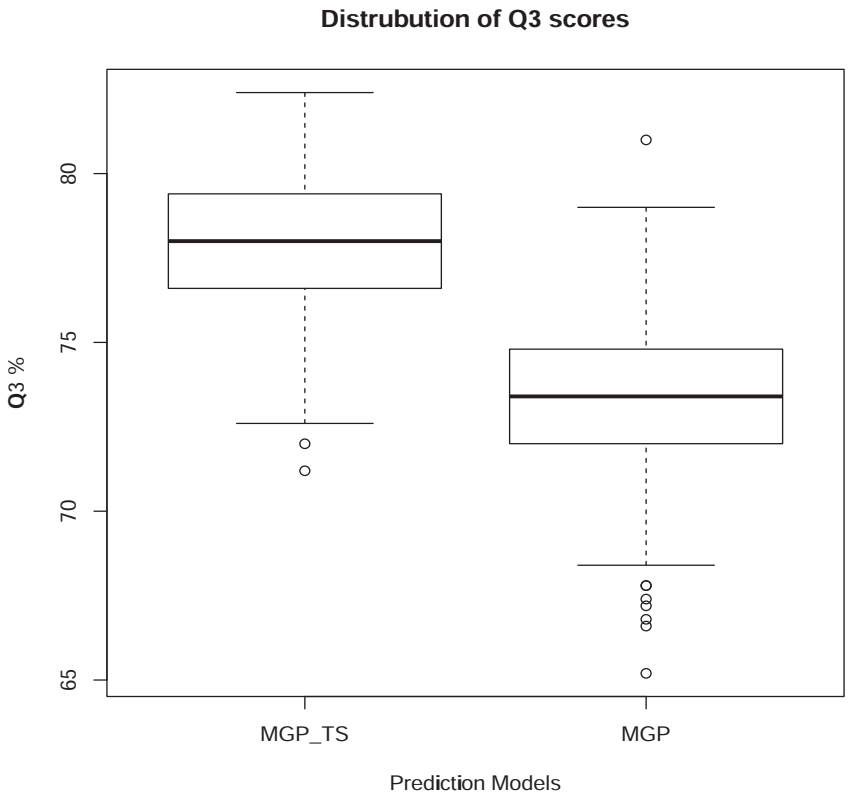

Figure 5.10: The performance of the proposed PSS prediction model using transition sites (MGP_TS), and without transition sites (MGP) on dataset PISCES.

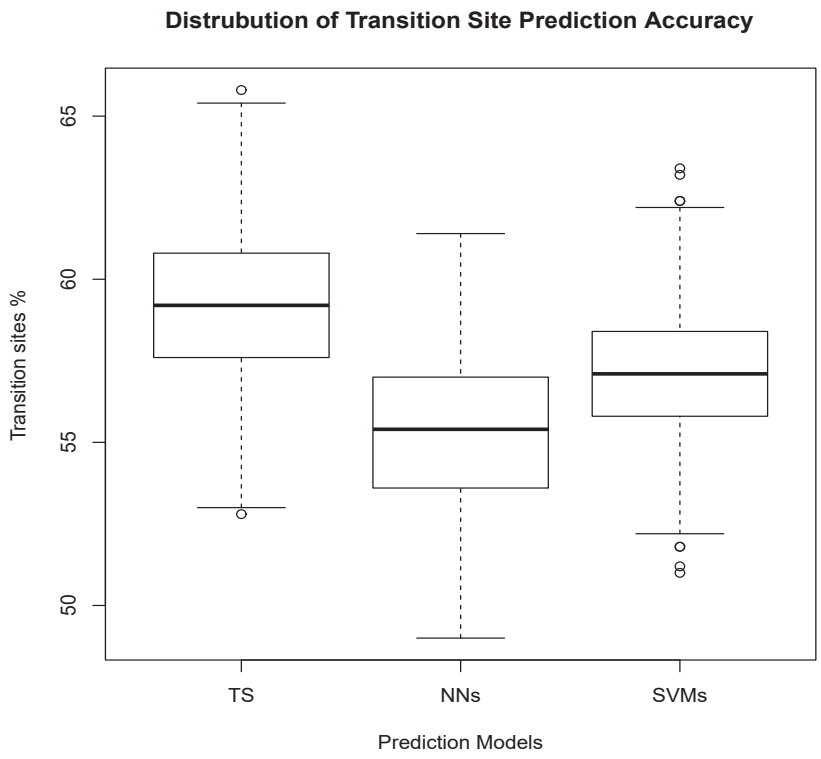

Figure 5.11: PSS transition sites (TS) prediction: by transition site (TS) model, and from the predicted secondary structures of two-tier ANNs and SVMs.

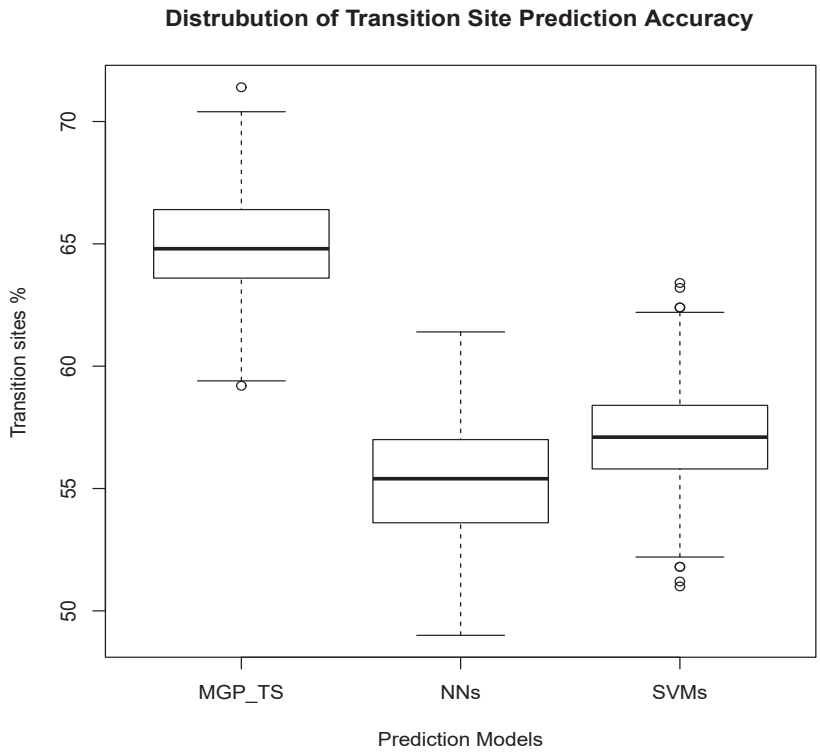

Figure 5.12: PSS transition sites (TS) prediction identified from the predicted secondary structures of the proposed multi-stage model with the transition site component (MGP_TS), and two-tier ANNs and SVMs.

Moreover, the performance of PSS transition site (TS) prediction model is compared with those of the proposed PSS classifier and the cascaded ANN and SVM classifiers in two steps: (1) PSS transition sites are identified from the predicted secondary structures of the three classifiers, and (2) the identified transition sites are compared to those directly predicted by the TS prediction model as shown in Figures 5.11 and 5.12. The distribution of transition site scores predicted directly by the TS model is approximately 3% and 4% higher than those identified from the PSS prediction of the two-tier SVM and ANN classifiers respectively, and the p-values of the statistical tests are less than 0.001 by the Wilcoxon rank-sum test as shown in Figure 5.11. The distribution of prediction scores from the experimental results does not follow a normal distribution when it is examined by using the Shapiro-Wilk normality test (Shapiro & Wilk, 1965) as was mentioned earlier. In addition, the distribution of transition site scores identified from the PSS prediction of the proposed two-stage PSS prediction model increases approximately 9% with the p-value that is less than 0.001 due to the improvement in the performance of the prediction model when it incorporates the information of transition sites as shown in Figure 5.12. The summary of all statistical results, including the distribution of Q_3 and transition site (TS) scores, are shown in Tables 5.13 and 5.14.

5.5.1 Pairwise Sequence Similarity in Protein Datasets

As we explained previously, the pairwise identity of protein sequences is an important criterion that should be considered for the learning and performance comparison of PSS prediction techniques. In other words, the selected PSS prediction models should incorporate at least a set of protein sequences with the same level of similarity. In order to investigate the effect of identity criterion, we compared the prediction performance of a two-tier ANN model which incorporates separately two protein datasets

Table 5.13: The distribution summary of Q_3 and transition site (TS) scores. Lower and upper whiskers denoted by LW and UW. Lower and upper hinges denoted by LH and UH respectively. Median is denoted by MED. Number of lower and upper outliers denoted by #OL.

#	Prediction Model	Score Type	LW	LH	MED	UH	UW	#OL
1	MGP	Q_3%	68.4	72.0	73.4	74.8	79.0	7, 1
2	NNs	Q_3%	65.0	68.4	69.6	70.8	74.2	9, 8
3	SVMs	Q_3%	66.2	70.0	71.2	72.6	76.0	4, 2
4	MGP_TS	Q_3%	72.6	76.6	78.0	79.4	82.4	2, 0
5	TS	TS%	53.0	57.6	59.2	60.8	65.4	2, 1
6	NNs	TS%	49.0	53.6	55.4	57.0	61.4	0, 0
7	SVMs	TS%	52.2	55.8	57.1	58.4	62.2	4, 3
8	MGP_TS	TS%	59.4	63.6	64.8	66.4	70.4	1, 1

Table 5.14: The summary of Wilcoxon statistical tests. The null hypothesis (H_0) is that the protein secondary structure (PSS) or transition site (TS) prediction distribution of the two tested prediction models are similar. Otherwise, the prediction distribution of the first prediction model is on the right of the prediction distribution of the other prediction method. Letter "X" indicates H_0 is rejected for the corresponding statistical test.

#	Prediction Models	Prediction Type	Wilcoxon Test	
			H_0	p-value
1	MGP & NNs	PSS	X	$< 10^{-3}$
2	MGP & SVMs	PSS	X	$< 10^{-3}$
3	MGP_TS & NNs	PSS	X	$< 10^{-3}$
4	MGP_TS & SVMs	PSS	X	$< 10^{-3}$
5	MGP_TS & MGP	PSS	X	$< 10^{-3}$
6	TS & NNs	TS	X	$< 10^{-3}$
7	TS & SVMs	TS	X	$< 10^{-3}$

from a protein culling server (Wang & Dunbrack, 2003) that contain sequences with 25% and 50% identity. The performance comparison of the prediction model with two datasets that have different level of homology are shown in Figure 5.13. The experimental results indicate the median of Q_3 scores' distribution is approximately 3% higher than that of the same prediction model when it incorporates a dataset with lower identity.

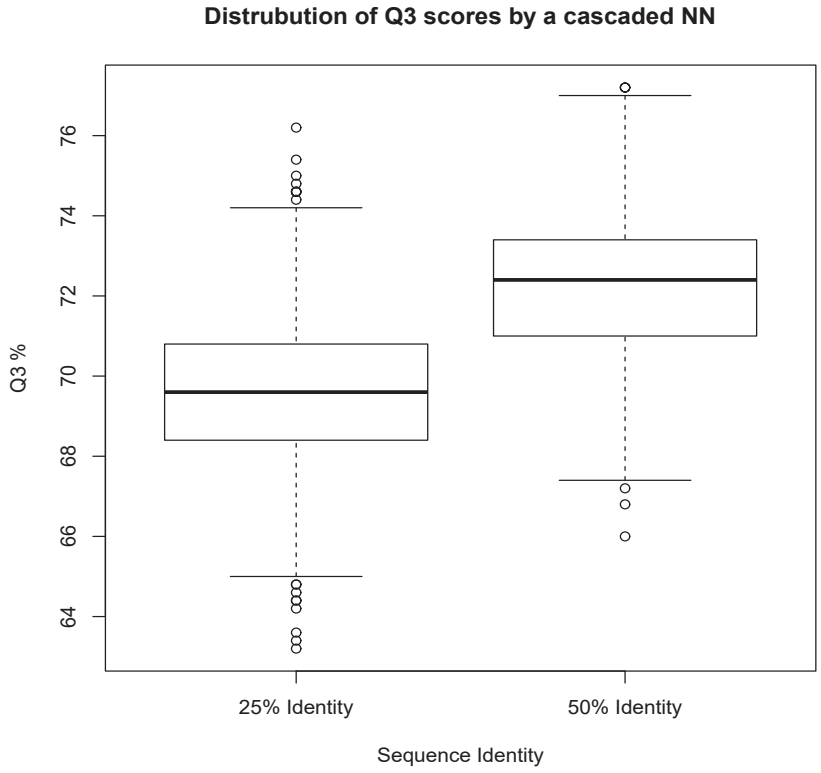

Figure 5.13: The distribution of Q_3 scores derived from different level of sequence identity.

Lastly, in this study, we used a large nonhomologous dataset and performed statistical tests on the experimental results. We strongly recommend the approach for a comprehensive performance evaluation in PSS prediction unlike using the conven-

tional cross-validation technique and datasets with relatively small sizes. The experimental results and statistical analyses indicate significant improvements in the performance of the proposed PSS prediction model compared to those of the cascaded ML techniques which commonly employ ANNs and SVMs. The information of PSS transition sites represent the topological property of protein backbones, and they can be also utilized for protein structure determination based on homology modeling and computational techniques. Moreover, we investigated the impact of a protein dataset's similarity on the outcome of PSS prediction, and the identity criterion should be considered while comparing the performances of PSS prediction methods. The proposed PSS prediction model can be further examined by employing different ANN architectures, protein profiles and PSS assignment schemes.

Chapter 6

Conclusions and Future work

6.1 Protein Structure Determination

Proteins are made up from the twenty types of amino acids which are connected by peptide bonds in linear orders. A polypeptide chain was described in four categories as the primary, secondary, tertiary and quaternary, after the structure of the first globular protein was elucidated (Linderstrøm-Lang, Boyer, Lardy, & Myrbäck, 1959). It was demonstrated that a protein's primary structure contains all the structural information to fold into a unique three-dimensional structure (Anfinsen et al., 1973).

In addition, in many biochemical experiments, it has been observed that many vital biological functions such as transport and storage, immune protection, coordinated motion, enzymatic catalysis which are carried out by proteins are very closely related to their unique tertiary structures (Laskowski, Watson, & Thornton, 2003; Travers, 1989; Bjorkman & Parham, 1990). Therefore, resolving proteins' tertiary structures from primary structures has potentially a large number of important applications including in synthesizing enzymes, drug designs for treatments, and agriculture (Chou, 2004; Noble et al., 2004).

At the present time, the most accurate protein structure determinations are experimental methods such as X-ray crystallography (Bragg et al., 1975; Blundell & Johnson, 1976) and nuclear magnetic resonance (NMR) spectroscopy (Wuthrich, 1986; Baldwin et al., 1991). However, these mainly experimental techniques are costly, time consuming and require very complex procedures (Metfessel & Saurugger, 1993; Johnson et al., 1994).

The percentage of proteins whose tertiary structures have been resolved experimentally to sequenced proteins has reached currently to less than 0.2%. This gap is increased constantly by identifying new protein sequences in a very high multitude due to rapid advances in large sequencing projects such as the Human Genome Project (Fleischmann, Adams, White, Clayton et al., 1995). Meanwhile, it has been shown that the number of unique structural folds from protein whose structures have been verified by experimental methods have been significantly declining in the last seven years.

Moreover, as alternative approaches for protein structure determination, a number of computational methods have been proposed with regards to the capabilities of the computational methods, and significant progress has been achieved in the computational fields in the last two decades. However, protein structure determination from amino acid sequences has remained the most fundamental problem in computational biology, and despite progress that has been made, protein structure prediction by computational methods has not reached a satisfactory level due to the extremely large input space of protein sequences.

6.2 Protein Secondary Structure Prediction

Protein secondary structures are linearly connected folds by which a protein structure can be represented with less detail while preserving the important part of the structural information. Protein secondary structure prediction methods have been utilized in protein structure prediction to reduce the complexities of tertiary structure prediction. Protein secondary structures are formed at the early stage of the protein folding pathway, and the secondary structure elements have decisive roles in the native conformation of a protein (Goldenberg et al., 1989).

In addition, the other advantage of using the information of protein secondary structures is evidenced by the fact that protein structures are more preserved than amino acid sequences since there are many nonhomologous proteins with similar structures. The preservation of structures is the basis upon which protein secondary structure prediction has served as an important complementary stage in a number of bioinformatic programs ranging from distant homology detections to multi-sequence alignments (Ginalski et al., 2003; Simossis & Heringa, 2005; Hargbo & Elofsson, 1999; Zhou & Zhou, 2005).

In addition, protein secondary structure prediction is used in protein structure determination methods based on protein threading and comparative modeling by which the correct identification of fold templates is the core functionality of the methods (Fischer & Eisenberg, 1996; Koretke et al., 1999; Zhou & Zhou, 2004; Sułkowska et al., 2012; Biasini et al., 2014). Moreover, the applications of accurately predicted secondary structures have improved significantly protein structure determination in *de novo* protein structure determination methods (Skolnick et al., 1997; Jones, 1999b; Wang et al., 2015). Protein secondary structure prediction methods are grouped in four classes as follows:

1. Statistical.

2. Based on stereo- and physico-chemical properties.

3. Based on sequence homology.

4. Machine Learning (ML).

In the early PSS prediction methods based on statistical methods, the likelihood of residue pairs in specific secondary structures are computed to quantify the interactions of the residue pairs within short ranges (Nagano, 1973; Chou & Fasman, 1974a). Moreover, other statistical approaches commonly are based on the techniques that formulate the correlation between sequence compositions and the different types of secondary structures by using Bayesian statistics and Information theory (Garnier et al., 1978; Kabsch et al., 1983; Gibrat et al., 1987; Garnier et al., 1996; Kloczkowski et al., 2002).

In PSS prediction methods based on stereo- and physico-chemical characteristics, a number of extensive prediction rules are derived by using the frequencies of secondary structures observed in proteins with known structures, and basically the principles that govern secondary structure formations are expressed by the prediction rules (Lim, 1974a; Ptitsyn & Finkelstein, 1983; Tanaka & Scheraga, 1976).

In PSS prediction methods based on sequence homology, the secondary structures of a query segment are estimated generally from the segment templates with known tertiary structures which have the highest similarity with the query segment, and the similarity of segment pairs are measured by using a substitution matrix. These methods are based on the speculation that the short segments of protein sequences with low similarity can possibly have similar secondary structures (Levin et al., 1986; Nishikawa & Ooi, 1986).

The latest PSS prediction methods are based on ML techniques which employ artificial neural networks, support vector machines and hidden markov models (Qian & Sejnowski, 1988; Asai et al., 1993; Hua & Sun, 2001). ML-based PSS prediction methods lead to higher prediction performances than those of statistical and sequence homology methods, and three-state prediction accuracies have reached to above 70% by adopting cascaded architectures and incorporating protein evolutionary information. By the alternative approaches of PSS prediction methods which cascaded architectures, hybrid designs and consensus schemes have adopted, the prediction accuracies have been improved further, and the reported prediction accuracies range from 70% to slightly above 80%.

In PSS prediction methods, there are several criteria which make a comparison challenging. The criteria that should be considered for a performance comparison are the similarity of datasets, the architectures and designs of PSS prediction methods, evaluation schemes, the types of input information, dataset sizes, generalization techniques and statistical hypothesis testing which have been explained further in the previous chapters. The prediction accuracy of a PSS prediction method cannot be used simply as a basis to compare the performance of the method with another PSS prediction method because the aforementioned criteria should be matched, at least to some degree, between the two prediction methods for an unbiased comparison. Otherwise, the PSS prediction methods are compared with different restrictions and experimental setups, and such a comparison may not be throughly informative about the actual performances of the two chosen methods as it was discussed in Section 5.5.1.

Lastly, encoding amino acid sequences from alphabetic representation to numeric values is an important step in many bioinformatic problems including PSS prediction since the information of sequences should be preserved after the encoding. For a protein sequence, using direct encoding each residue is assigned to a numeric multi-

dimensional vector which preserves the arrangement of local residues, whereas in indirect encoding the global frequencies of two or more adjacent residues are preserved.

6.3 Experiments and Summary of Results

In this thesis, we developed several techniques, specifically: protein protein encoding schemes, a PSS transition site prediction method, and a PSS prediction model. A series of experiments have been performed in progressive steps to examine the performances of the proposed techniques separately and evaluate thoroughly the overall performance when the proposed techniques are aggregated as a PSS prediction system. The summary of the contributions of this work, the proposed techniques, and the experiments performed in this thesis are categorized as follows.

6.3.1 Codon Encoding Scheme

Initially, we developed the novel *codon* encoding scheme which is a direct amino acid encoding scheme based on the genetic codon mappings. Next, we introduced a generalized technique to evaluate amino acid encoding schemes which commonly are used in PSS prediction methods based on ML techniques. The codon encoding scheme has been compared to fourteen encoding techniques with different schemes and dimensions by using the proposed evaluation technique. The codon encoding that we developed outperformed the other encoding schemes for learning five commonly used substitution matrices with regards to the learning speed and accuracy (Zamani & Kremer, 2011)[“Amino acid encoding schemes for machine learning methods”].

6.3.2 Secondary Structure Prediction and Codon Encoding

In addition, the performance and effectiveness of the codon encoding scheme have been compared to those of the commonly used *orthogonal* encoding when the two encoding techniques are used for secondary structure prediction. In the study, we constructed a number of binary and tertiary support vector machines (SVMs), evaluated the performances of the two encoding schemes, and compared them to the prediction results of several studies that employed similar PSS prediction models based on SVMs.

The prediction models that incorporated the codon encoding scheme outperformed the PSS prediction models that incorporated the orthogonal encoding, and in some cases comparable prediction results with those PSS prediction models that incorporated protein profiles. The outcomes reconfirmed the effectiveness of the new generalized technique that we developed in Section 6.3.1 for the evaluation of amino acid encoding schemes (Zamani & Kremer, 2012)["Protein Secondary Structure Prediction Using Support Vector Machines and a Codon Encoding Scheme"].

6.3.3 An Evolutionary PSS Model using Clustering

In this study, we also proposed a new protein secondary structure prediction method based on a genetic programming (GP) technique. The evolutionary PSS model incorporates a new encoding technique which we developed based on the k-means clustering technique. The performance of the GP model is compared to those of feed-forward artificial neural networks (ANNs) and support vector machines (SVMs). The performances of the three prediction methods were examined by two sets of experiments. Initially, the prediction methods were represented by sequence information. Then, the three prediction methods incorporated the statistical information derived from clus-

tered sequences. The protein sequences were encoded by using the codon encoding scheme prior to the clustering.

According to the experimental results, our novel GP-based prediction model increased the three-state prediction accuracy approximately from 2% to 3% compared to the other two prediction models employing ANNs and SVMs. Also, the new prediction model incorporated statistical information performed on average 3% higher than the similar prediction models using amino acid sequences. Moreover, the number of input units is reduced if the statistical information derived from clusters are represented to the three prediction models (Zamani & Kremer, 2015b)["Protein secondary structure prediction using an evolutionary computation method and clustering"].

6.3.4 Two-stage PSS Model and Score Vectors

In this experiment, we extended the proposed GP-based PSS prediction method to a new two-stage PSS prediction model which is cascaded with ANNs. In the first stage, encoded protein sequences are clustered, and protein profiles are mapped to regions which are defined based on a Ramachandran map. Then, the identified regions and statistical information derived from clustered sequences are formulated by a novel approach to construct *score vectors* based on Information theory.

The score vectors are incorporated in the second stage for predicting secondary structures. By using score vectors like we have done in this work, fewer input units are required in comparison to incorporating protein profiles. Therefore, our score vector scheme reduces the number of learning parameters and computational complexity in the second stage. The performance of the two-stage prediction model has been compared to the state-of-the-art PSS prediction method based on ML techniques which commonly employ cascaded feed-forward ANNs and SVMs. The experimental results indicated on an average 3% higher prediction accuracy for our new method compared

to those of the current best approaches using cascaded ANNs and SVMs (Zamani & Kremer, 2015a)["A multi-stage protein secondary structure prediction system using machine learning and information theory"].

6.3.5 Transition Site Framework and PSS Prediction

In the last experiments, we developed the two-stage PSS prediction model explained in Section 6.3.4 through a novel framework of PSS transition sites. Transition sites are the locations on a protein backbone where one secondary structure element ends and another secondary structure element starts. In the first stage, we extended a second tier of a feed-forward ANN to perform transition site prediction. The information of determined transition sites are added to their corresponding score vectors, and PSS prediction is performed in the second stage.

The performance of our two-stage prediction model with PSS transition sites is compared to the state-of-the-art ML methods which commonly employ cascaded ANNs and SVMs. The evaluation is performed in two steps: (1) PSS transition site prediction, (2) PSS structure prediction. The statistical analyses were performed on the experimental results indicate significant improvements. The distribution of three-state prediction scores by the two-stage prediction model without using PSS transition sites is approximately 3.5% and 2.5% higher than those of the two-tier ANN and SVM predictors respectively.

The distribution of the three-state scores is increased an overall 6% with p-values less than 0.001 when our two-stage prediction model incorporates the information of PSS transition sites. Also, the distribution of transition site scores identified from the predicted secondary structures of the two-stage prediction model increased approximately 9% with p-values less than 0.001 compared to those of the other (previously top performing) prediction models (Zamani & Kremer, 2016)["Protein secondary struc-

ture prediction through a novel framework of secondary structure transition sites and new encoding schemes"].

6.3.6 Protein Dataset's Similarity

As discussed in the previous chapters, two important criteria for comparing the performances of two PSS prediction methods are the similarity of protein datasets and the evaluation technique used for the prediction outcomes. In Section 5.5.1, we investigated the effect of two different levels of datasets' similarity on the performance of a PSS prediction method. In addition, we used a very large dataset which contains protein sequences with pairwise identity of less than 25%, and the dataset is the most recent compilation from PISCES (Wang & Dunbrack, 2003).

The commonly used evaluation of PSS prediction is based on computing the average correlation coefficient of the three secondary structures by using a cross-validation technique. In our experiments, we initially measured and compared the performances of the PSS prediction methods by the common aforementioned evaluation technique. However, we also extended our tests beyond the normal standard and performed, additionally, a number of statistical tests and analyses based on a Wilcoxon rank-sum nonparametric test (Wilcoxon et al., 1970) for a more comprehensive comparison of the proposed prediction model with the other PSS predictors. It is important to note that the level of statistical rigor performed in this study has not been employed previously in the field of PSS prediction. Our proposed evaluation approach based on the statistical tests and analyses distinguishes the distributions of prediction scores achieved by different PSS prediction methods in a statistically sound manner used in most other domains where performance comparisons are required. By showing that it is possible to apply these statistical techniques and obtain statistically meaningful results, and we have set a new standard for performance evaluation which we hope

will be adopted for a more comprehensive evaluation of PSS prediction methods in other studies.

6.4 Contribution

Protein structure determination from amino acid sequences is one of the most complex problems in computational biology, and it has a number of important applications such as synthesizing enzymes and drug designs for treatments. Any computational technique such as a robust protein secondary structure prediction method that can be utilized as complementary tools with the aim of reducing the complexity, costs and time of protein structure determination is crucially important. Protein structure prediction can be also incorporated in homology detection and multi-sequence alignment tools since protein structures which are represented by less complexity as secondary structures are more preserved than protein sequences.

In this thesis, we developed a new evaluation framework to examine the performance of an amino acid encoding scheme which is an important step in solving protein structures in methods based on Machine Learning approaches (Section 6.3.1). In addition, we developed two new amino acid encoding schemes based on the genetic codon mapping, clustering, information theory and Ramachandran map, and the performance improvements of incorporating the encoding schemes have been examined in PSS prediction (Sections 6.3.1, 6.3.2, and 6.3.3). Moreover, in this study, we developed a protein secondary structure prediction method that performs better than the previous methods. The experimental results and statistical analyses indicate our novel two-stage PSS prediction model derives useful information from protein sequences and profiles, and performs statistically better than the leading state-of-the-art approaches. In addition, protein secondary structure transition sites represent

the topology of protein backbones, and transition site prediction can be utilized in comparative modeling or computational methods of protein tertiary structure determination to identify the boundary of secondary structure elements (Sections 6.3.4 and 6.3.5).

In addition, we have developed a testing and evaluation methodology that is more meaningful in terms of both generalization to non-homologous proteins and also statistical rigour. It is recommended that our new evaluation approach be applied in all future protein secondary structure studies for a more comprehensive comparison and evaluation which is based on the distributions of the prediction scores (Section 6.3.5 and 6.3.6).

6.5 Future work

The performance of the proposed protein secondary structure prediction model can be improved potentially by constructing hybrid schemes that consist of different machine learning techniques and data representations. The two-stage secondary structure prediction model can be examined with more complex neural network architectures such as Recurrent Neural Networks and Deep Neural Networks. The prediction model can be incorporated with other protein profiles such as PSI-BLAST, and different schemes of protein secondary structure assignments, GORV and JNET. In addition, designing a parallel scheme for the proposed GP technique can reduce the computational complexity and speed up the learning process.

Bibliography

Adamczak, R., Porollo, A., & Meller, J. (2005). Combining prediction of secondary structure and solvent accessibility in proteins. *Proteins: Structure, Function, and Bioinformatics*, *59*(3), 467–475.

Altschul, S., Gish, W., Miller, W., Myers, E., Lipman, D., et al. (1990). Basic local alignment search tool. *Journal of molecular biology*, *215*(3), 403–410.

Altschul, S., Madden, T., Schäffer, A., Zhang, J., Zhang, Z., Miller, W., & Lipman, D. (1997). Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, *25*(17), 3389–3402.

Anfinsen, C., et al. (1973). Principles that govern the folding of protein chains. *Science*, *181*(96), 223–230.

Asai, K., Hayamizu, S., & Handa, K. (1993). Prediction of protein secondary structure by the hidden markov model. *Computer applications in the biosciences: CABIOS*, *9*(2), 141–146.

Babaei, S., Geranmayeh, A., & Seyyedsalehi, S. A. (2010). Protein secondary structure prediction using modular reciprocal bidirectional recurrent neural networks. *Computer methods and programs in biomedicine*, *100*(3), 237–247.

Bairoch, A., Apweiler, R., Wu, C., Barker, W., Boeckmann, B., Ferro, S., Gasteiger,

E., Huang, H., Lopez, R., Magrane, M., et al. (2005). The universal protein resource (uniprot). *Nucleic acids research*, *33*(suppl 1), D154–D159.

Baldi, P., & Brunak, S. (2001). Bioinformatics: The machine learning approach. In *Bioinformatics: The Machine Learning Approach*. MIT Press.

Baldi, P., Brunak, S., Frasconi, P., Soda, G., & Pollastri, G. (1999). Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics*, *15*(11), 937–946.

Baldwin, E. T., Weber, I. T., St Charles, R., Xuan, J.-C., Appella, E., Yamada, M., Matsushima, K., Edwards, B., Clore, G. M., & Gronenborn, A. M. (1991). Crystal structure of interleukin 8: symbiosis of nmr and crystallography. *Proceedings of the National Academy of Sciences*, *88*(2), 502–506.

Barto, A. G., Sutton, R. S., & Anderson, C. W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *Systems, Man and Cybernetics, IEEE Transactions on*, *13*(5), 834–846.

Bengio, Y. (2009). Learning deep architectures for ai. *Machine Learning*, *2*(1), 1–127.

Berman, H., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., Shindyalov, I., & Bourne, P. (2000). The protein data bank. *Nucleic acids research*, *28*(1), 235–242.

Bettella, F., Rasinski, D., & Knapp, E. W. (2012). Protein secondary structure prediction with sparrow. *Journal of chemical information and modeling*, *52*(2), 545–556.

Biasini, M., Bienert, S., Waterhouse, A., Arnold, K., Studer, G., Schmidt, T., Kiefer, F., Cassarino, T. G., Bertoni, M., Bordoli, L., et al. (2014). Swiss-model: modelling

protein tertiary and quaternary structure using evolutionary information. *Nucleic acids research*, (p. gku340).

Bingru, Y., Wei, H., Zhun, Z., & Huabin, Q. (2009). Kaapro: An approach of protein secondary structure prediction based on kdd in the compound pyramid prediction model. *Expert Systems with Applications*, *36*(5), 9000–9006.

Bjorkman, P. J., & Parham, P. (1990). Structure, function, and diversity of class i major histocompatibility complex molecules. *Annual Review of Biochemistry*, *59*, 253–288.

Blundell, T., Bedarkar, S., Rinderknecht, E., & Humbel, R. (1978). Insulin-like growth factor: a model for tertiary structure accounting for immunoreactivity and receptor binding. *Proceedings of the National Academy of Sciences*, *75*(1), 180–184.

Blundell, T., & Johnson, L. (1976). Protein crystallography. *Molecular Biology Series*.

Bowie, J., Luthy, R., & Eisenberg, D. (1991). A method to identify protein sequences that fold into a known three-dimensional structure. *Science*, *253*(5016), 164–170.

Bradley, P., Misura, K., & Baker, D. (2005). Toward high-resolution de novo structure prediction for small proteins. *Science*, *309*(5742), 1868–1871.

Bragg, W. L., Phillips, D. C., Lipson, H., et al. (1975). *Development of X-ray analysis*. G. Bell.

Brameier, M., & Banzhaf, W. (2001). A comparison of linear genetic programming and neural networks in medical data mining. *Evolutionary Computation, IEEE Transactions on*, *5*(1), 17–26.

Burkowski, F. (2008). *Structural bioinformatics: an algorithmic approach*. Chapman & Hall/CRC.

Chambers, D., & Mandic, J. (2001). Recurrent neural networks for prediction: learning algorithms architecture and stability. *John Wiley & Sons, Ltd., Chichester*, *18*, 32.

Chang, C., & Lin, C. (2011). Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *2*(3), 27.

Chen, C., Chen, L., Zou, X., & Cai, P. (2009). Prediction of protein secondary structure content by using the concept of chou's pseudo amino acid composition and support vector machine. *Protein and peptide letters*, *16*(1), 27–31.

Chen, J., & Chaudhari, N. S. (2007). Cascaded bidirectional recurrent neural networks for protein secondary structure prediction. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, *4*(4), 572–582.

Chothial, C., & Lesk, A. M. (1986). The relation between the divergence of sequence and structure in proteins. *The EMBO Journal*, *5*(4), 823–826.

Chou, K.-C. (2004). Structural bioinformatics and its impact to biomedical science. *Current Medicinal Chemistry*, *11*, 2105–2134.

Chou, P., & Fasman, G. (1974a). Conformational parameters for amino acids in helical, beta-sheet, and random coil regions calculated from proteins. *Biochemistry*, *13*(2), 211–222.

Chou, P., & Fasman, G. (1974b). Prediction of protein conformation. *Biochemistry*, *13*(2), 222–245.

Chou, P., Fasman, G., et al. (1978). Prediction of the secondary structure of proteins from their amino acid sequence. *Adv Enzymol Relat Areas Mol Biol*, *47*(2), 45–148.

Cireşan, D., & Meier, U. (2015). Multi-column deep neural networks for offline handwritten chinese character classification. In *2015 International Joint Conference on Neural Networks (IJCNN)*, (pp. 1–6). IEEE.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273–297.

Crammer, K., & Singer, Y. (2002). On the learnability and design of output codes for multiclass problems. *Machine Learning*, *47*(2), 201–233.

Crick, F., et al. (1970). Central dogma of molecular biology. *Nature*, *227*(5258), 561–563.

Cuff, J., & Barton, G. (1999). Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*, *34*(4), 508–519.

Cuff, J., Barton, G., et al. (2000). Application of multiple sequence alignment profiles to improve protein secondary structure prediction. *Proteins Structure Function and Genetics*, *40*(3), 502–511.

Dayhoff, M., Schwartz, R., & Orcutt, B. (1978). A model of evolutionary change in proteins. *In Atlas of Protein Sequences and Structure*, *5*, 345–352.

Dongardive, J., & Abraham, S. (2015). Secondary structure prediction of protein using resilient back propagation learning algorithm. *BRAIN. Broad Research in Artificial Intelligence and Neuroscience*, *6*(1-2), 22–29.

Doolittle, R. F. (1981). Similar amino acid sequences: chance or common ancestry. *Science*, *214*(4517), 149–159.

Doolittle, R. F. (1986). *Of URFs and ORFs: A primer on how to analyze derived amino acid sequences*. University Science Books.

Dor, O., & Zhou, Y. (2007). Achieving 80% ten-fold cross-validated accuracy for secondary structure prediction by large-scale training. *PROTEINS: Structure, Function, and Bioinformatics*, *66*, 838–845.

Eggermont, J., Eiben, A. E., & van Hemert, J. I. (1999). A comparison of genetic programming variants for data classification. In *International Symposium on Intelligent Data Analysis*, (pp. 281–290). Springer.

Faraggi, E., Zhang, T., Yang, Y., Kurgan, L., & Zhou, Y. (2012). Spine x: improving protein secondary structure prediction by multistep learning coupled with prediction of solvent accessible surface area and backbone torsion angles. *Journal of computational chemistry*, *33*(3), 259–267.

Finkelstein, A., & Ptitsyn, O. (1971). Statistical analysis of the correlation among amino acid residues in helical, beta-structural and non-regular regions of globular proteins. *Journal of molecular biology*, *62*(3), 613–624.

Fischer, D., & Eisenberg, D. (1996). Protein fold recognition using sequence-derived predictions. *Protein Science*, *5*(5), 947–955.

Fleischmann, R. D., Adams, M. D., White, O., Clayton, R. A., et al. (1995). Whole-genome random sequencing and assembly of haemophilus influenzae rd. *Science*, *269*(5223), 496–512.

Fodje, M., & Al-Karadaghi, S. (2002). Occurrence, conformational features and amino acid propensities for the π-helix. *Protein engineering*, *15*(5), 353–358.

Frishman, D., & Argos, P. (2004). Knowledge-based protein secondary structure assignment. *Proteins: Structure, Function, and Bioinformatics*, *23*(4), 566–579.

Garnier, J., Gibrat, J., Robson, B., et al. (1996). Gor method for predicting protein secondary structure from amino acid sequence. *Methods in enzymology*, *266*, 540–553.

Garnier, J., Osguthorpe, D., Robson, B., et al. (1978). Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *Journal of molecular biology*, *120*(1), 97.

Gathercole, C., & Ross, P. (1994). Dynamic training subset selection for supervised learning in genetic programming. In *International Conference on Parallel Problem Solving from Nature*, (pp. 312–321). Springer.

Gibrat, J., Garnier, J., & Robson, B. (1987). Further developments of protein secondary structure prediction using information theory: new parameters and consideration of residue pairs. *Journal of molecular biology*, *198*(3), 425–443.

Ginalski, K., Pas, J., Wyrwicz, L. S., Von Grotthuss, M., Bujnicki, J. M., & Rychlewski, L. (2003). Orfeus: detection of distant homology using sequence profiles and predicted secondary structure. *Nucleic acids research*, *31*(13), 3804–3807.

Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. *Foundations of genetic algorithms*, *1*, 69–93.

Goldenberg, D. P., Frieden, R. W., Haack, J. A., & Morrison, T. B. (1989). Mutational analysis of a protein-folding pathway. *Nature*, *338*, 127–132.

Green, J. R., Korenberg, M. J., & Aboul-Magd, M. O. (2009). Pci-ss: Miso dynamic nonlinear protein secondary structure prediction. *BMC bioinformatics*, *10*(1), 1.

Guha, S., Rastogi, R., & Shim, K. (2001). Cure: an efficient clustering algorithm for large databases. *Information Systems*, *26*(1), 35–58.

Guo, J., Chen, H., Sun, Z., & Lin, Y. (2004). A novel method for protein secondary structure prediction using dual-layer svm and profiles. *PROTEINS: Structure, Function, and Bioinformatics*, *54*, 738–743.

Hargbo, J., & Elofsson, A. (1999). Hidden markov models that use predicted secondary structures for fold recognition. *Proteins: Structure, Function, and Bioinformatics*, *36*(1), 68–76.

Hartigan, J. A., & Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *28*(1), 100–108.

Hecht-Nielsen, R. (1989). Self-organization and associative memory. *IEEE J. Quant. Electron*, *25*(2), 237.

Henikoff, S., & Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, *89*(22), 10915–10919.

Holland, J. H. (1992). Genetic algorithms. *Scientific American*, *267*(1), 66–72.

Holley, L., & Karplus, M. (1989). Protein secondary structure prediction with a neural network. *Proceedings of the National Academy of Sciences*, *86*(1), 152–156.

Holm, L., & Sander, C. (1996). Mapping the protein universe. *Science*, *273*(5275), 595.

Hu, H., Pan, Y., Harrison, R., & Tai, P. (2004). Improved protein secondary structure prediction using support vector machine with a new encoding scheme and an advanced tertiary classifier. *NanoBioscience, IEEE Transactions on*, *3*(4), 265–271.

Hua, S., & Sun, Z. (2001). A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach1. *Journal of molecular biology*, *308*(2), 397–407.

Hubbard, T. J., Murzin, A. G., Brenner, S. E., & Chothia, C. (1997). Scop: a structural classification of proteins database. *Nucleic acids research*, *25*(1), 236–239.

Johnson, M., Srinivasan, N., Sowdhamini, R., & Blundell, T. (1994). Knowledge-based protein modeling. *Critical Reviews in Biochemistry and Molecular Biology*, *29*(1), 1–68.

Jones, D. (1999a). Protein secondary structure prediction based on position-specific scoring matrices. *Journal of molecular biology*, *292*(2), 195–202.

Jones, D. T. (1999b). Genthreader: an efficient and reliable protein fold recognition method for genomic sequences. *Journal of molecular biology*, *287*(4), 797–815.

Kabsch, W., & Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, *22*(12), 2577–2637.

Kabsch, W., Sander, C., et al. (1983). How good are predictions of protein secondary structure? *FEBS letters*, *155*(2), 179–182.

Karplus, K., Sjölander, K., Barrett, C., Cline, M., Haussler, D., Hughey, R., Holm, L., Sander, C., et al. (1997). Predicting protein structure using hidden markov models. *Proteins Structure Function and Genetics*, *29*(s 1), 134–139.

Kawashima, S., & Kanehisa, M. (2000). Aaindex: amino acid index database. *Nucleic acids research*, *28*(1), 374–374.

Kim, H., & Park, H. (2003). Protein secondary structure prediction based on an improved support vector machines approach. *Protein Engineering*, *16*(8), 553–560.

King, S., & Johnson, W. (1999). Assigning secondary structure from protein coordinate data. *Proteins: Structure, Function, and Bioinformatics*, *35*(3), 313–320.

Kleywegt, G. J., & Jones, T. A. (1996). Phi/psi-chology: Ramachandran revisited. *Structure*, *4*(12), 1395–1400.

Kloczkowski, A., Ting, K., Jernigan, R., & Garnier, J. (2002). Combining the gor v algorithm with evolutionary information for protein secondary structure prediction from amino acid sequence. *Proteins: Structure, Function, and Bioinformatics*, *49*(2), 154–166.

Kneller, D., Cohen, F., & Langridge, R. (1990). Improvement in protein secondary structure prediction by an enhanced neural network. *Journal of molecular biology*, *214*(1), 171–182.

Koh, I., Eyrich, V., Marti-Renom, M., Przybylski, D., Madhusudhan, M., Eswar, N., Grana, O., Pazos, F., Valencia, A., Sali, A., et al. (2003). Eva: evaluation of protein structure prediction servers. *Nucleic acids research*, *31*(13), 3311–3315.

Koretke, K. K., Russell, R. B., Copley, R. R., & Lupas, A. N. (1999). Fold recognition using sequence and secondary structure information. *Proteins: Structure, Function, and Bioinformatics*, *37*(S3), 141–148.

Kountouris, P., & Hirst, J. D. (2009). Prediction of backbone dihedral angles and protein secondary structure using support vector machines. *BMC bioinformatics*, *10*(1), 1.

Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection*, vol. 1. MIT press.

Labesse, G., Colloc'h, N., Pothier, J., & Mornon, J. (1997). P-sea: A new efficient

assignment of secondary structure from cα trace of proteins. *Computer applications in the biosciences: CABIOS*, *13*(3), 291–295.

Lac, H., & Kremer, S. (2009). *Inducing fold dynamics from known protein structures using machine learning*. Ph.D. thesis, CIS, University of Guelph.

Laskowski, R. A., Watson, J. D., & Thornton, J. M. (2003). From protein structure to biochemical function? *Journal of structural and functional genomics*, *4*(2-3), 167–177.

Leman, J. K., Mueller, R., Karakas, M., Woetzel, N., & Meiler, J. (2013). Simultaneous prediction of protein secondary structure and transmembrane spans. *Proteins: Structure, Function, and Bioinformatics*, *81*(7), 1127–1140.

Levin, J., Pascarella, S., Argos, P., & Garnier, J. (1993). Quantification of secondary structure prediction improvement using multiple alignments. *Protein Engineering*, *6*(8), 849–854.

Levin, J., Robson, B., & Garnier, J. (1986). An algorithm for secondary structure determination in proteins based on sequence similarity. *FEBS letters*, *205*(2), 303–308.

Li, Z., & Yu, Y. (2016). Protein secondary structure prediction using cascaded convolutional and recurrent neural networks. *arXiv preprint arXiv:1604.07176*.

Lichtarge, O., Bourne, H., Cohen, F., et al. (1996). An evolutionary trace method defines binding surfaces common to protein families. *Journal of molecular biology*, *257*(2), 342–358.

Lim, V. (1974a). Algorithms for prediction of α-helical and β-structural regions in globular proteins. *Journal of Molecular Biology*, *88*(4), 873–894.

Lim, V. (1974b). Structural principles of the globular organization of protein chains. a stereochemical theory of globular protein secondary structure. *Journal of molecular biology*, *88*(4), 857–872.

Lin, K., Simossis, V., Taylor, W., & Heringa, J. (2005). A simple and fast secondary structure prediction method using hidden neural networks. *Bioinformatics*, *21*(2), 152–159.

Linderstrøm-Lang, K., Boyer, J. S. P., Lardy, H., & Myrbäck, K. (1959). *The Enzymes*, vol. 1. Academic Press, New York.

Loveard, T., & Ciesielski, V. (2001). Representing classification problems in genetic programming. In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, vol. 2, (pp. 1070–1077). IEEE.

Madera, M., Calmus, R., Thiltgen, G., Karplus, K., & Gough, J. (2010). Improving protein secondary structure prediction using a simple k-mer model. *Bioinformatics*, *26*(5), 596–602.

Maetschke, S., Towsey, M., & Bodén, M. (2005). Blomap: An encoding of amino acids witch improves signal peptide cleavage site prediction. In *Proceedings of the third Asia Pacific bioinformatics conference*, vol. 1, (pp. 141–150). Singapore: Imperial College Press.

Magnan, C. N., & Baldi, P. (2014). Sspro/accpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity. *Bioinformatics*, *30*(18), 2592–2597.

Martin, J., Gibrat, J., & Rodolphe, F. (2005a). Choosing the optimal hidden markov model for secondary-structure prediction. *Intelligent Systems, IEEE*, *20*(6), 19–25.

Martin, J., Gibrat, J., & Rodolphe, F. (2006). Analysis of an optimal hidden markov model for secondary structure prediction. *BMC structural biology*, *6*(1), 25–45.

Martin, J., Letellier, G., Marin, A., Taly, J., De Brevern, A., & Gibrat, J. (2005b). Protein secondary structure assignment revisited: a detailed analysis of different assignment methods. *BMC Structural Biology*, *5*(1), 17.

Matthews, B. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, *405*(2), 442–451.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*(4), 115–133.

Metfessel, B. A., & Saurugger, P. N. (1993). Pattern recognition in the prediction of protein structural class. In *System Sciences, 1993, Proceeding of the Twenty-Sixth Hawaii International Conference on*, vol. 1, (pp. 679–688). IEEE.

Mirabello, C., & Pollastri, G. (2013). Porter, paleale 4.0: high-accuracy prediction of protein secondary structure and relative solvent accessibility. *Bioinformatics*, *29*(16), 2056–2058.

Montgomerie, S., Sundararaj, S., Gallin, W., & Wishart, D. (2006). Improving the accuracy of protein secondary structure prediction using structural alignment. *BMC Bioinformatics*, *7*(1), 1–13.

Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T., & Tramontano, A. (2016). Critical assessment of methods of protein structure prediction: Progress and new directions in round xi. *Proteins: Structure, Function, and Bioinformatics*.

Moult, J., Hubbard, T., Bryant, S., Fidelis, K., & Pedersen, J. (1998). Critical

assessment of methods of protein structure prediction (casp): Round ii. *Proteins: Structure, Function, and Bioinformatics*, *29*(S1), 2–6.

Moult, J., Pedersen, J., Judson, R., & Fidelis, K. (2004). A large-scale experiment to assess protein structure prediction methods. *Proteins: Structure, Function, and Bioinformatics*, *23*(3), ii–iv.

Nagano, K. (1973). Logical analysis of the mechanism of protein folding: I. predictions of helices, loops and β-structures from primary structure. *Journal of molecular biology*, *75*(2), 401–420.

Needleman, S. B., & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, *48*(3), 443–453.

Nguyen, M., Rajapakse, J., et al. (2003). Multi-class support vector machines for protein secondary structure prediction. *Genome Informatics Series*, (pp. 218–227).

Nishikawa, K., & Ooi, T. (1986). Amino acid sequence homology applied to the prediction of protein secondary structures, and joint prediction with existing methods. *Biochimica et Biophysica Acta (BBA)-Protein Structure and Molecular Enzymology*, *871*(1), 45–54.

Noble, M. E., Endicott, J. A., & Johnson, L. N. (2004). Protein kinase inhibitors: insights into drug design from structure. *Science*, *303*(5665), 1800–1805.

Oltean, M., & Dioşan, L. (2009). An autonomous gp-based system for regression and classification problems. *Applied Soft Computing*, *9*(1), 49–60.

Pauling, L., & Corey, R. (1951a). The pleated sheet, a new layer configuration of polypeptide chains. *Proceedings of the National Academy of Sciences of the United States of America*, *37*(5), 251.

Pauling, L., & Corey, R. B. (1951b). Configurations of polypeptide chains with favored orientations around single bonds two new pleated sheets. *Proceedings of the National Academy of Sciences*, *37*(11), 729–740.

Pauling, L., Corey, R. B., & Branson, H. R. (1951). The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. *Proceedings of the National Academy of Sciences*, *37*(4), 205–211.

Periti, P., Quagliarotti, G., & Liquori, A. (1967). Recognition of alpha-helical segments in proteins of known primary structure. *Journal of molecular biology*, *24*(2), 313.

Pollastri, G., & Mclysaght, A. (2005). Porter: a new, accurate server for protein secondary structure prediction. *Bioinformatics*, *21*(8), 1719–1720.

Pollastri, G., Przybylski, D., Rost, B., & Baldi, P. (2002). Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins: Structure, Function, and Bioinformatics*, *47*(2), 228–235.

Prothero, J. (1966). Correlation between the distribution of amino acids and alpha helices. *Biophysical Journal*, *6*(3), 367.

Przybylski, D., & Rost, B. (2002). Alignments grow, secondary structure prediction improves. *Proteins: Structure, Function, and Bioinformatics*, *46*(2), 197–205.

Ptitsyn, O., & Finkelstein, A. (1983). Theory of protein secondary structure and algorithm of its prediction. *Biopolymers*, *22*(1), 15–25.

Qian, N., & Sejnowski, T. (1988). Predicting the secondary structure of globular proteins using neural network models. *Journal of molecular biology*, *202*(4), 865–884.

Qu, W., Sui, H., Yang, B., & Qian, W. (2011). Improving protein secondary structure prediction using a multi-modal bp method. *Computers in biology and Medicine*, *41*(10), 946–959.

Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.

Ramachandran, G., & Sasisekharan, V. (1968). Conformation of polypeptides and proteins. *Adv Protein Chem*, *23*(283), 438.

Robson, B. (1974). Analysis of the code relating sequence to conformation in globular proteins. theory and application of expected information. *Biochemical Journal*, *141*(3), 853–867.

Robson, B., & Suzuki, E. (1976). Conformational properties of amino acid residues in globular proteins. *Journal of molecular biology*, *107*(3), 327–356.

Rost, B. (1996). Phd: predicting 1d protein structure by profile based neural networks. *Meth. Enzymol*, *266*, 525–539.

Rost, B. (1997). Protein structures sustain evolutionary drift. *Folding & Design*, *2*, 519–524.

Rost, B. (1999). Twilight zone of protein sequence alignments. *Protein engineering*, *12*(2), 85–94.

Rost, B., & Sander, C. (1994a). Combining evolutionary information and neural networks to predict protein secondary structure. *Proteins: Structure, Function, and Bioinformatics*, *19*(1), 55–72.

Rost, B., & Sander, C. (1994b). Conservation and prediction of solvent accessibility in protein families. *PROTEINS: Structure, Function, and Genetics*, *20*, 216–226.

Rost, B., & Sander, C. (2000). Third generation prediction of secondary structures. *Protein structure prediction: Methods and protocols*, (pp. 71–95).

Rost, B., Sander, C., & Schneider, R. (1994). Redefining the goals of protein secondary structure prediction. *Journal of molecular biology*, *235*(1), 13–26.

Rost, B., Sander, C., et al. (1993). Prediction of protein secondary structure at better than 70% accuracy. *Journal of molecular biology*, *232*(2), 584–599.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.

Salamov, A., & Solovyev, V. (1997). Protein secondary structure prediction using local alignments. *Journal of molecular biology*, *268*(1), 31–36.

Salamov, A., Solovyev, V., et al. (1995). Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments. *Journal of Molecular Biology*, *247*(1), 11–15.

Sander, C., & Schneider, R. (1991). Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins: Structure, Function, and Bioinformatics*, *9*(1), 56–68.

Sasagawa, F., & Tajima, K. (1993). Prediction of protein secondary structures by a neural network. *Computer applications in the biosciences: CABIOS*, *9*(2), 147–152.

Schneider, R. (1989). Secondary structure prediction of proteins, including any tertiary structure aspects. *Department of Biology, Univ. Heidelberg, FRG, Diploma thesis*.

Schölkopf, B., Burges, C., Vapnik, V., Uthurusamy, F. R., et al. (1995). Extracting support data for a given task. In *First International Conference on Knowledge Discovery Data Mining (KDD-95)*, (pp. 252–257). AAAI Press.

Shannon, C., Weaver, W., Blahut, R., & Hajek, B. (1949). *The mathematical theory of communication*, vol. 117. University of Illinois press Urbana.

Shapiro, S. S., & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, *52*(3/4), 591–611.

Simossis, V. A., & Heringa, J. (2005). Praline: a multiple sequence alignment toolbox that integrates homology-extended and secondary structure information. *Nucleic acids research*, *33*(suppl 2), W289–W294.

Skolnick, J., Kolinski, A., & Ortiz, A. R. (1997). Monsster: a method for folding globular proteins with a small number of distance restraints. *Journal of molecular biology*, *265*(2), 217–241.

Smith, T. F., & Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of molecular biology*, *147*(1), 195–197.

Solovyev, V., & Salamov, A. (1994). Predicting α-helix and β-strand segments of globular proteins. *Computer applications in the biosciences: CABIOS*, *10*(6), 661–669.

Spencer, M., Eickholt, J., & Cheng, J. (2015). A deep learning network approach to ab initio protein secondary structure prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, *12*(1), 103–112.

Sułkowska, J. I., Morcos, F., Weigt, M., Hwa, T., & Onuchic, J. N. (2012). Genomics-aided structure prediction. *Proceedings of the National Academy of Sciences*, *109*(26), 10340–10345.

Sussman, J., Lin, D., Jiang, J., Manning, N., Prilusky, J., Ritter, O., & Abola, E. (1998). Protein data bank (pdb): database of three-dimensional structural infor-

mation of biological macromolecules. *Acta Crystallographica Section D: Biological Crystallography*, *54*(6), 1078–1084.

Swanson, R. (1984). A vector representation for amino acid sequences. *Bulletin of mathematical biology*, *46*(4), 623–639.

Szent-Gyorgyi, A., Cohen, C., et al. (1957). Role of proline in polypeptide chain configuration of proteins. *Science (New York, NY)*, *126*(3276), 697.

Tanaka, S., & Scheraga, H. A. (1976). Medium-and long-range interaction parameters between amino acids for predicting three-dimensional structures of proteins. *Macromolecules*, *9*, 945–950.

Taylor, W. R. (1986). The classification of amino acid conservation. *Journal of theoretical Biology*, *119*(2), 205–218.

Travers, A. (1989). Dna conformation and protein binding. *Annual review of biochemistry*, *58*, 427.

Tsochantaridis, I., Joachims, T., Hofmann, T., & Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *J. Mach. Learn. Res.*, *6*, 1453–1484.

Vapnik, V., & Chapelle, O. (2000). Bounds on error expectation for support vector machines. *Neural computation*, *12*(9), 2013–2036.

Vivarelli, F., Giusti, G., Villani, M., Campanini, R., Fariselli, P., Compiani, M., & Casadio, R. (1995). Lgann: a parallel system combining a local genetic algorithm and neural networks for the prediction of secondary structure of proteins. *Computer applications in the biosciences: CABIOS*, *11*(3), 253–260.

Wang, G., & Dunbrack, R. (2003). Pisces: a protein sequence culling server. *Bioinformatics*, *19*(12), 1589–1591.

Wang, R. Y.-R., Kudryashev, M., Li, X., Egelman, E. H., Basler, M., Cheng, Y., Baker, D., & DiMaio, F. (2015). De novo protein structure determination from near-atomic-resolution cryo-em maps. *Nature methods*, *12*(4), 335–338.

Wang, S., Peng, J., Ma, J., & Xu, J. (2016). Protein secondary structure prediction using deep convolutional neural fields. *Scientific reports*, *6*.

Wang, Z., Zhao, F., Peng, J., & Xu, J. (2011). Protein 8-class secondary structure prediction using conditional neural fields. *Proteomics*, *11*(19), 3786–3792.

Wei, Y., Thompson, J., & Floudas, C. (2012). Concord: a consensus method for protein secondary structure prediction via mixed integer linear optimization. *Proceedings of the Royal Society of London Series A: Mathematical, Physical and Engineering Sciences*, *468*, 831–850.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, *78*(10), 1550–1560.

Weston, J., & Watkins, C. (1999). Support vector machines for multi-class pattern recognition. In *Proceedings of the Seventh European Symposium on Artificial Neural Networks (ESANN 99)*, vol. 4, (pp. 219–224).

Wilcoxon, F., Katti, S., & Wilcox, R. A. (1970). Critical values and probability levels for the Wilcoxon rank sum test and the Wilcoxon signed rank test. *Selected tables in mathematical statistics*, *1*, 171–259.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, *1*(1), 67–82.

Won, K., Hamelryck, T., Prügel-Bennett, A., & Krogh, A. (2007). An evolutionary method for learning HMM structure: prediction of protein secondary structure. *BMC bioinformatics*, *8*(1), 357.

Won, K., Prügel-Bennett, A., & Krogh, A. (2004). Training HMM structure with genetic algorithm for biological sequence analysis. *Bioinformatics*, *20*(18), 3613–3619.

Wu, C., & McLarty, J. (2000). *Neural networks and genome informatics*, vol. 1. Elsevier Science.

Wu, C., Whitson, G., McLarty, J., Ermongkonchai, A., Chang, T., et al. (1992). Protein classification artificial neural system. *Protein science: a publication of the Protein Society*, *1*(5), 667–677.

Wuthrich, K. (1986). *NMR of proteins and nucleic acids*. Wiley.

Yi, T., & Lander, E. (1993). Protein secondary structure prediction using nearest-neighbor methods. *Journal of molecular biology*, *232*(4), 1117–1129.

Zahn, C. T. (1971). Graph-theoretical methods for detecting and describing gestalt clusters. *Computers, IEEE Transactions on*, *100*(1), 68–86.

Zamani, M., & Kremer, S. (2011). Amino acid encoding schemes for machine learning methods. In *Bioinformatics and Biomedicine Workshops (BIBMW), 2011 IEEE International Conference on*, (pp. 327–333). IEEE.

Zamani, M., & Kremer, S. (2012). Protein secondary structure prediction using support vector machines and a codon encoding scheme. In *Bioinformatics and Biomedicine Workshops (BIBMW), 2012 IEEE International Conference on*, (pp. 22–27). IEEE.

Zamani, M., & Kremer, S. C. (2015a). A multi-stage protein secondary structure prediction system using machine learning and information theory. In *Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on*, (pp. 1304–1309). IEEE.

Zamani, M., & Kremer, S. C. (2015b). Protein secondary structure prediction using an evolutionary computation method and clustering. In *Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2015 IEEE Conference on*, (pp. 1–6). IEEE.

Zamani, M., & Kremer, S. C. (2016). Protein secondary structure prediction through a novel framework of secondary structure transition sites and new encoding schemes. In *Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2016 IEEE Conference on*. IEEE (accepted).

Zemla, A., Venclovas, Č., Fidelis, K., & Rost, B. (1999). A modified definition of sov, a segment-based measure for protein secondary structure prediction assessment. *Proteins: Structure, Function, and Bioinformatics*, *34*(2), 220–223.

Zhang, X., Mesirov, J., Waltz, D., & Cohen, F. (1992). Hybrid system for protein secondary structure prediction. *Journal of molecular biology*, *225*(4), 1049–1063.

Zhou, H., & Zhou, Y. (2004). Single-body residue-level knowledge-based energy score combined with sequence-profile and secondary structure information for fold recognition. *Proteins: Structure, Function, and Bioinformatics*, *55*(4), 1005–1013.

Zhou, H., & Zhou, Y. (2005). Spem: improving multiple sequence alignment with sequence profiles and predicted secondary structures. *Bioinformatics*, *21*(18), 3615–3621.

Zhou, J., & Troyanskaya, O. G. (2014). Deep supervised and convolutional generative stochastic network for protein secondary structure prediction. *arXiv preprint arXiv:1403.1347*.

Zimmermann, K., & Gibrat, J.-F. (2010). Amino acid "little big bang": Representing amino acid substitution matrices as dot products of euclidian vectors. *BMC bioinformatics*, *11*(1), 1.

Zimmermann, O., & Hansmann, U. (2008). Locustra: accurate prediction of local protein structure using a two-layer support vector machine approach. *Journal of chemical information and modeling*, *48*(9), 1903–1908.

Zuckerkandl, E., & Pauling, L. (1965). Evolutionary divergence and convergence in proteins. *Evolving genes and proteins*, *97*, 97–166.

Zvelebil, M. J., Barton, G. J., Taylor, W. R., & Sternberg, M. J. (1987). Prediction of protein secondary structure and active sites using the alignment of homologous sequences. *Journal of molecular biology*, *195*(4), 957–961.