Fast learning optimized prediction methodology
for protein secondary structure prediction,
relative solvent accessibility prediction and
phosphorylation prediction

by

Saraswathi Sundararajan

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Major: Bioinformatics and Computational Biology

Program of Study Committee:
Robert L. Jernigan, Co-major Professor
Richard Honzatko, Co-major Professor
Andrzej Kloczkowski
Drena Dobbs
Alicia Carriquiry
David Fernández-Baca

Iowa State University

Ames, Iowa

2011

DEDICATION

I would like to dedicate this thesis to ....

Maatha, Phitha , Guru and Deivam.....

Table of Contents

List of Tables

List of Figures

ABREVIATIONS

	C-α	Alpha carbon in an amino acid residue
	C-β	Beta carbon in an amino acid residue
	CABS	C-α -C-β-Side group protein model
	CB513	Cuff and Barton dataset (Cuff and Barton, 2000)
	DSSP	Dictionary of protein secondary structure
	ELM	Extreme Learning Machine
	GA	Genetic Algorithm
	H, E, C	α-helix, β-sheet and Coil
	HSSP	Homology-derived Secondary Structure of Proteins
	MSA	Multiple Sequence Alignment
	NH_2	Amino terminus or N terminus of an amino acid
	NMR	Nuclear Magnetic Resonance
	PCA	Principal Component Analysis
	PDB	Protein Data Bank
	PSO	Particle Swarm Optimization
	PSRP	Position Specific Residue Preferences of amino acids
	Q_3	The overall accuracy for all SSP
	RS126	Rost and Sander data set
	RSCBPDB	Newer name for Protein Data Bank
	S-ELM	Sparse-ELM
	SOV	Segment Overlap Measure
	SSP	Secondary Structure Prediction
	Datasets - used	
1	dataset-84	84 out of CB513 proteins - initial study
2	dataset-415	415 out of CB513 proteins - larger study
3	Aaindex dataset	544 physicochemical properties of amino acids
4	PSRP dataset	Position Specific Residue Preference data
5	Manesh-dataset	215 proteins for RSA prediction
6	Phosphoto. ELM database	Has 13, 604 sequences

ACKNOWLEDGEMENTS

I would like to thank my major professors Prof. Jernigan, Prof. Kloczkowski and Prof. Honzatko for guiding me in my research and my POS committee members Prof. Drena Dobbs, Prof. Alicia Carriquiry, Prof. Dimitris Margaritis and Prof. David Fernandez-Baca for their support and guidance in my research. I have had the most wonderful time as a student in the Jernigan lab. We have the kind of gentle mentorship that inspires without pressure and instills confidence and hope to achieve great heights with respect to our research work. Prof. Jernigan and Prof. Kloczkowski, have been my immediate supervisors and have provided me with unflinching support needed for the successful completion of my studies. I consider myself very fortunate for having had the opportunity to work with them, since they made me feel welcome and proud of my achievements.

I met Prof. Drena Dobbs, my IGERT PI soon after coming to Iowa State. She has been my mentor and given me invaluable guidance and support throughout my studies. Her enthusiastic support enabled me to attend many conferences and meet eminent researchers and collaborators who inspired me to pursue my research with increasing enthusiasm. Trish Stauble, my program coordinator, was my very first contact here in the BCB program. There are some friendships that are destined to be lifelong relationships, that become dearer by the day. My friendship with Trish and Drena belong to this kind. Trish and Drena are the kind of people whom people would like to meet wherever they go. They bring so much enthusiasm and assistance to the students of the BCB program that *they almost make us believe* this is an easy thing to do.

I would like to thank professor Juan Luis Fernández Martínez from the University of Oviedo, Spain, whom I met at the IJCCI conference in Spain last year and who has helped

me to advance my research. He has become my mentor, collaborator and friend in the short time that I have known him.

I am especially thankful to the Bioinformatics and Computational Biology program for giving me the opportunity to pursue my PhD studies here at Iowa State. I am very grateful for the financial support provided for my research by the Bioinformatics and Computational Biology program, the IGERT program (IGERT-0504304) and National Institutes of Health grants R01GM081680, R01GM072014, and R01GM073095.

I would like to express my thanks to my mentors at Iowa State. My introduction to Iowa state and the BCB program came in 2006, when I came across a book in Computational Molecular Biology, edited by Prof. Srinivas Aluru. I have known Prof. Srinivas Aluru and Prof. Chris Tuggle since my admission to the BCB program and they have been pillars of support during my studies here. I had many discussions with them, which set the pace for my research in the BCB program. I would like to thank Prof. Marit Nilsen-Hamilton whose thoughtful guidance gave me some insights in my cancer research and helped me publish my very first journal publication. I would like to thank Prof. Allen Miller, in whose lab I did my first rotation and was introduced to the intricacies of wet lab experiments by his lab manager, Randy Beckett. I would like to thank Prof. Vasant Honovar in whose lab I did my second rotation and whose thoughtful advice has helped me with my research since. I could not have had such a wonderful time here if it were not for the support of these wonderful people who believed in me.

I would like to remember my roots which brought me to Iowa State. I started my research in Bioinformatics at Amrita University (AU), Coimbatore, India in 2004 and later continued it at Nanyang Technological University (NTU), Singapore during my MS studies in Bioinformatics. I would like to thank my advisors and professors at Amrita University and NTU, under whose guidance I started my research in Bioinformatics.

I would like to take this opportunity to express my thanks to the wonderful people, who are my friends and classmates, who helped me find a second home here at Iowa State. I would like to thank my fellow BCB students and my seniors who provided early support as

mentors. I would like to thank the staff members at ISU, BBMB, BCB and the Jernigan lab for all the assistance provided to me during my studies. I am thankful that I found such a place to fulfill my dreams and am honored to be one of the students in the bioinformatics and computational biology program, here at Iowa State. Last but not least, I would like to thank the members of my family who wished me success in my endeavors.

ABSTRACT

Computational methods are rapidly gaining importance in the field of structural biology, mostly due to the explosive progress in genome sequencing projects and the large disparity between the number of sequences and the number of structures. There has been an exponential growth in the number of available protein sequences and a slower growth in the number of structures. There is therefore an urgent need to develop computed structures and identify the functions of these sequences. Developing methods that will satisfy these needs both efficiently and accurately is of paramount importance for advances in many biomedical fields, for a better basic understanding of aberrant states of stress and disease, including drug discovery and discovery of biomarkers.

Several aspects of secondary structure predictions and other protein structure-related predictions are investigated using different types of information such as data obtained from knowledge-based potentials derived from amino acids in protein sequences, physicochemical properties of amino acids and propensities of amino acids to appear at the ends of secondary structures. Investigating the performance of these secondary structure predictions by type of amino acid highlights some interesting aspects relating to the influences of the individual amino acid types on formation of secondary structures and points toward ways to make further gains. Other research areas include Relative Solvent Accessibility (**RSA**) predictions and predictions of phosphorylation sites, which is one of the Post-Translational Modification (**PTM**) sites in proteins.

Protein secondary structures and other features of proteins are predicted efficiently, reliably, less expensively and more accurately. A novel method called Fast Learning Optimized PREDiction (**FLOPRED**) Methodology is proposed for predicting protein secondary struc-

tures and other features, using knowledge-based potentials, a Neural Network based Extreme Learning Machine (**ELM**) and advanced Particle Swarm Optimization (**PSO**) techniques that yield better and faster convergence to produce more accurate results. These techniques yield superior classification of secondary structures, with a training accuracy of 93.33% and a testing accuracy of 92.24% with a standard deviation of 0.48% obtained for a small group of 84 proteins. We have a Matthew's correlation-coefficient ranging between 80.58% and 84.30% for these secondary structures. Accuracies for individual amino acids range between 83% and 92% with an average standard deviation between 0.3% and 2.9% for the 20 amino acids. On a larger set of 415 proteins, we obtain a testing accuracy of 86.5% with a standard deviation of 1.38%. These results are significantly higher than those found in the literature.

Prediction of protein secondary structure based on amino acid sequence is a common technique used to predict its 3-D structure. Additional information such as the biophysical properties of the amino acids can help improve the results of secondary structure prediction. A database of protein physicochemical properties is used as features to encode protein sequences and this data is used for secondary structure prediction using **FLOPRED**. Preliminary studies using a Genetic Algorithm (**GA**) for feature selection, Principal Component Analysis (**PCA**) for feature reduction and **FLOPRED** for classification give promising results.

Some amino acids appear more often at the ends of secondary structures than others. A preliminary study has indicated that secondary structure accuracy can be improved as much as 6% by including these effects for those residues present at the ends of α-helix, β-strand and coil.

A study on **RSA** prediction using **ELM** shows large gains in processing speed compared to using support vector machines for classification. This indicates that **ELM** yields a distinct advantage in terms of processing speed and performance for **RSA**. Additional gains in accuracies are possible when the more advanced **FLOPRED** algorithm and **PSO** optimization are implemented.

Phosphorylation is a post-translational modification on proteins often controls and regulates their activities. It is an important mechanism for regulation. Phosphorylated sites are

known to be present often in intrinsically disordered regions of proteins lacking unique tertiary structures, and thus less information is available about the structures of phosphorylated sites. It is important to be able to computationally predict phosphorylation sites in protein sequences obtained from mass-scale sequencing of genomes. Phosphorylation sites may aid in the determination of the functions of a protein and to better understanding the mechanisms of protein functions in healthy and diseased states. **FLOPRED** is used to model and predict experimentally determined phosphorylation sites in protein sequences. Our new **PSO** optimization included in **FLOPRED** enable the prediction of phosphorylation sites with higher accuracy and with better generalization. Our preliminary studies on 984 sequences demonstrate that this model can predict phosphorylation sites with a training accuracy of 92.53% , a testing accuracy 91.42% and Matthew's correlation coefficient of 83.9%.

In summary, secondary structure prediction, Relative Solvent Accessibility and phosphorylation site prediction have been carried out on multiple sets of data, encoded with a variety of information drawn from proteins and the physicochemical properties of their constituent amino acids. Improved and efficient algorithms called **S-ELM** and **FLOPRED**, which are based on Neural Networks and Particle Swarm Optimization are used for classifying and predicting protein sequences. Analysis of the results of these studies provide new and interesting insights into the influence of amino acids on secondary structure prediction. **S-ELM** and **FLOPRED** have also proven to be robust and efficient for predicting relative solvent accessibility of proteins and phosphorylation sites. These studies show that our method is robust and resilient and can be applied for a variety of purposes. It can be expected to yield higher classification accuracy and better generalization performance compared to previous methods.

PART I

GENERAL INTRODUCTION

CHAPTER 1. INTRODUCTION

1.1 Motivation

Computational methods are rapidly gaining importance in the field of structural biology, mostly due to the explosive progress in genome sequencing projects. This has resulted in exponential growth in the number of available protein sequences with large numbers having unknown structures and unknown functions. There is an urgent need to identify the structure and function of these sequences with higher efficiency and accuracy. Developing methods that will satisfy these needs is of paramount importance for advances in many biomedical fields, including drug discovery and discovery of biomarkers, for a better basic understanding of aberrant states of stress and disease.

Proteins are the essence of all living beings. They perform a variety of biological functions and are essential for the well being of their hosts. Malfunctioning or unfolding of proteins can result in various diseases in humans and other organisms. Knowledge of protein functions can help attain various goals such as:

- Better medical care and quality of life for humans.
- Conduct studies for drug development and better understanding of genomes that impact health and well being.
- Conduct studies in genetic engineering of plants and animals for productivity and manufacture of safe industrial products that might impact humans and the environment.

Various factors influence protein functions, such as a protein's native structure, information coded in its constituent amino acid sequences and its interactions with the surrounding

solvent environment which is influenced by the Relative Solvent Accessibility **(RSA)** values of its residues. Methods which can predict protein structure and other properties which impact protein interactions, such as relative solvent accessibility and phosphorylation, occupy a central important role in structural biological research.

1.2 Specific aims of this study

The primary goal of our studies has been to predict protein secondary structures and other features of protein sequences, such as relative solvent accessibility and phosphorylation, *more accurately, efficiently and more reliably* compared to existing algorithms found in literature. Considering the large amount of data that needs to be processed, another important goal was to develop an algorithm which would be *cost effective, simple and fast,* which will run without large demands on resources and provide prediction results within a reasonable amount of time. To achieve these goals, a novel method called Fast Learning Optimized Prediction **(FLOPRED)** methodology is proposed for predicting protein secondary structure and other protein properties. Knowledge-based potentials and other sequence related information such as physicochemical properties of amino acids are encoded as features in the data set. **FLOPRED** methodology uses a modified version of a neural network algorithm called Extreme Learning Machine **(ELM)** which is extremely fast compared to a traditional neural network. The parameters of **ELM** are optimized using an advanced form of Particle Swarm Optimization (**PSO**).

Several aspects of secondary structure prediction and protein related features were investigated using different types of data as listed below:

- Secondary structure prediction using data derived from knowledge-based potentials in protein sequences and **FLOPRED** algorithm. We have been successful in achieving high testing accuracies exceeding 90%, as discussed in Section 4.4 on page 86
- Secondary structure prediction using a database of physical and chemical properties of amino acids **AAindex** (Kawashima et al., 1999). We find that secondary structures can

be predicted with testing accuracy of 70% using only these properties as features, as discussed in Section 6.3 on page 125.

- Study of secondary structure prediction using the propensity of amino acids to appear at the ends of secondary structures (Richardson and Barlow, 1999; Duan et al., 2008). We find that secondary structure prediction accuracies can be improved by at least 6% by using these propensities, as discussed in Section 7.6 on page 139.

- Prediction of Relative Solvent Accessibility (**RSA**) of proteins. Our results are comparable to others in the literature with testing accuracies between 60% and 89%. This study mainly illustrates the rapid speed of ELM, which is several times faster than existing algorithms, as discussed in Section 9.2 on page 161.

- Prediction of phosphorylation sites, which is one of the Post-Translational Modification (**PTM**) sites in proteins. We find that phosphorylation sites can be predicted with a testing accuracy of 91.42%, as discussed in Section 10.3 on page 173.

The **FLOPRED** method was used on each of the above aspects of protein sequences and properties, to build and develop probabilistic models for protein structure and function predictions. All the experiments that were performed gave results that were higher than found in literature for similar studies; indicating that the *Fast Learning Optimized Prediction methodology* (**FLOPRED**) is robust and reliable when applied to a variety of data, encoding a variety of protein amino acid content, properties and functions.

1.3 Contributions of this study

1.3.1 Highly accurate prediction of protein secondary structures

The main contribution of this work is the prediction of protein secondary structures *with higher accuracy and efficiency* compared to existing algorithms found in literature.

1.3.2 Development of an efficient computational methodology

One contribution of this work is the development of an efficient and robust computational methodology called **FLOPRED** to make predictions related to protein secondary structures and other protein features. In addition, **FLOPRED** is expected to be cost effective, simple to use, run without large demands on resources, provide prediction results within a reasonable amount of time and yield higher classification accuracies. The **FLOPRED** technique can have a wide variety of applications in bioinformatics and can be used to obtain higher accuracies in multi-class prediction and classification problems on a variety of data encoding protein or genetic information.

1.4 Knowledge-based methods used in this study

The various knowledge based methods used in this thesis are discussed below. A detailed description of the methods and optimization, such as **ELM-PSO, FLOPRED** and **PSO**, which are used throughout this study are discussed in this chapter. A description of other methods such as **GA** and **PCA** will be given when they are actually applied in the studies detailed in the following chapters. The methods used are as follows:

- An Extreme Learning Machine classifier **(ELM-PSO)** (Suresh et al., 2010; Saraswathi et al., 2011) based on neural networks. **FLOPRED** is an improved version of **(ELM-PSO)**.
- Particle Swarm Optimization **(PSO)** (Kennedy and Eberhart, 1995; Fernández-Martínez and García-Gonzalo, 2008) based on the natural behavior of individuals in groups.
- Genetic Algorithm **(GA)** (Goldberg, 1989) based on evolutionary search techniques.
- Principal Component Analysis **(PCA)** (Pearson, 1901; Fernández-Martínez et al., 2010) which performs an orthogonal decomposition of a given data to derive its principal uncorrelated components.

Figure 1.1 A single layer Neural Network

This figure shows a **SLFN** that has inputs that are mapped to outputs. We have one input layer followed by one hidden layer and an output layer. The input layer presents each amino acid, encoded as 27 knowledge-based potential features $(P_1, (P_2, (P_3, ...(P_{27})$, which are normalized to have values between 0 and 1. The hidden layer has weights (W_{11}) and biases (b_{11}), which are the parameters of the network. *Many such hidden layer units will be used where* 11 *indicates first layer, first weight, first bias and* 21 *indicates second layer, first weight* for the neural network. Any non-linear activation function can learn N distinct observations by tuning the input weights and the number of hidden neurons. Activation functions such as sigmoidal and Gaussian functions can be used for the hidden neuron layer which outputs $a = logsig(Wp+b)$, while a linear activation function is used for the output neurons. In the **ELM** network, *a second set of weights and biases are analytically calculated* using the Moore-Penrose inverse matrix. The calculated parameters are learned during training and stored for later use during testing. The output layer has three units (only one shown) which gives a vector of three real values (output), one for each of the three secondary structures. The maximum of these three values is considered to be the predicted structure for the residue of interest. In this figure the third value is the highest, and it will be assigned as coil (C).

1.4.1 Extreme Learning Machine classifier

The Extreme Learning Machine classifier called **ELM-PSO** consists of two units, which are the Single Layer Feedforward Network **(SLFN)** based Extreme Learning Machine (**ELM**) algorithm and the Particle Swarm Optimization (**PSO**) algorithm that is used to tune the parameters of the ELM.

1.4.1.1 Single Layer Feedforward Networks

Neural networks such as Single Layer Feedforward Networks (**SLFN**) have the capability of approximating an existing function which relates a set of inputs to the outputs, to within a

small error β. A **SLFN** with N hidden neurons and randomly chosen input weights can learn a network with N distinct observations to within a small error, (Huang and Babri, 1998). **SLFN** use slow gradient based methods for learning and tuning their parameters. An **SLFN** network can be trained on a finite set of data, which has at most N hidden neurons. Any non-linear activation function can learn N distinct observations with zero error. In theory, this is possible if the input weights that connect the input layer and the hidden layers can be adjusted or tuned for all these networks. These weights and the bias form the *parameters* of these **SLFN** (Figure 1.1). If these weights are iteratively adjusted using *traditional gradient based methods*, there will be a dependency between different layers of parameters. This can result in improper learning, convergence to local minima and the need for many more iterations to reach good generalization performance.

1.4.1.2 Extreme Learning Machine

ELM is a modified version of Single Layer Feed-forward Network (**SLFN**) where the input weights are chosen randomly and the output weights are calculated analytically. Activation functions such as sigmoidal and Gaussian functions can be used for the hidden neuron layer, while a linear activation function is used for the output neurons. **ELM** is a fast and simple algorithm compared to traditional Neural Networks and is capable of finding the best results using smaller resources. If the parameters of **SLFN** (input weights and the bias of the hidden layer) are randomly chosen, **SLFNs** *become a linear system* in which the output weights can be determined analytically through a Moore-Penrose generalized pseudo-inverse operation of the hidden layer output matrices. *This improved algorithm is called the Extreme Learning Machine*. A comprehensive overview of **ELM** was given by Huang et al., in (Huang et al., 2006).

ELM has better generalization performance since the norm of its weights is small (Huang et al., 2006). Theoretically, **ELM** speeds up computations considerably, providing for better generalization performance and enabling extremely fast speeds during processing. On comparative studies (Huang et al., 2006), **ELM** show that they can do as well or better than

traditional methods such as Support Vector Machines **(SVM)**, while enabling faster computations compared to other Feedforward networks. We perform a comparative study using a Sparse-ELM called **S-ELM** (Suresh et al., 2010) which gives much faster performance speed as compared to Support Vector Machines as discussed in 9.1 on page 157. The **S-ELM** is a modified form of **ELM** which works well for data that are imbalanced, where the number of features exceed the number of samples that are available for modeling. The features of the **ELM** can be summarized as having:

- The smallest training error.
- Smallest norm of weights.
- Best generalization performance.
- Extremely rapid convergence compared to other neural networks.

1.4.1.3 Optimization of Extreme Learning Machine

If the number of training samples N is equal to the number of hidden neurons then the network can approximate the training parameters with zero error. For very large data sets, however, it will be computationally intensive to use a large number of hidden neurons. Hence it is necessary to approximate the parameters to obtain outputs close to the observed solution with minimum error. So, to train a **SLFN** with fixed input weights W_i, bias b_i, and a single hidden layer, we only need to find the least squares solution that minimizes the error. The **ELM** algorithm minimizes this error by tuning its parameters using **PSO**, as explained next.

1.4.1.4 ELM-PSO algorithm

It has been shown that optimal selection of **ELM** parameters (input weights, bias values and hidden neurons) can minimize errors and Particle Swam Optimization (**PSO**) can give much improved prediction results (Suresh et al., 2010; Saraswathi et al., 2011). **ELM-PSO** consists of the Extreme Learning Machine (**ELM**) classifier as the main algorithm, which uses a set of training samples to build a model. The weights from the hidden layer to the output

layer are analytically calculated. During the training phase, the **PSO** is called upon to optimize the parameters, such as the weights, the number of hidden neurons and the biases of the **ELM**, which result in improved classification accuracy. These parameters are stored and used during the testing phase. *A simple* **PSO** algorithm is used for these initial studies.

1.4.1.5 Fast Learning Optimized Prediction methodology (FLOPRED)

The **ELM-PSO** was used on several data sets initially that yielded promising results. Later on, *an improved version of* **ELM-PSO**, named Fast Learning Optimized Prediction methodology (**FLOPRED**), which *combines the simplicity of* **ELM** *and powerful* **PSO** *algorithms with optimized and improved search techniques* was developed for advanced studies in our research. The **ELM** and the advanced **PSO** algorithms are described next.

1.4.1.6 Extreme Learning Machine algorithm

Let V be $H \times n$ input weights, b be $H \times 1$ bias values for each hidden neuron and W be $C \times H$ output weights, for a multi-category classification (C-distinct classes) problem. If we have N observations $(X_i, T_i, i = 1, 2, \ldots, N)$, the outputs of the **ELM** network with H hidden neurons can be defined as,

$$y_k = \sum_{j=1}^{H} w_{kj} G_j\left(V, b, X_i\right), \quad k = 1, 2, \cdots, C \tag{1.1}$$

where $G_j\left(.\right)$ is the output of the j^{th} hidden neuron and $G_j(.)$ is the activation function.

For sigmoidal hidden neurons, the output of j^{th} hidden neuron $G_j(.)$ is defined as

$$G_j(V, b, X_i) = \tanh\left(b_j + \sum_{k=1}^{N} v_{jk} x_i^k\right), \quad j = 1, 2, \cdots, H \tag{1.2}$$

In the case of a radial basis function (**RBF**), the output of the j^{th} Gaussian neuron $G_j(.)$ is defined as

$$G_j(V, b, X_i) = e^{\frac{-||X_i - V_j||}{2b_j^2}}, \quad j = 1, 2, \cdots, H \tag{1.3}$$

where b acts as the width of the Gaussian hidden neuron. Equation (1.1) can be written in matrix form as

$$\hat{Y} = W\ Y_h \tag{1.4}$$

where Y_h is a $HbyN$ matrix, which is defined as,

$$\mathbf{Y_h} = \begin{bmatrix} G_1(V,b,X_1) & G_1(V,b,X_2) & \cdots & G_1(V,b,X_N) \\ \vdots & \vdots & \vdots & \vdots \\ G_H(V,b,X_1) & G_H(V,b,X_2) & \cdots & G_H(V,b,X_N) \end{bmatrix} \tag{1.5}$$

The target (t_i^k) is defined as

$$t_i^k = \begin{cases} 1 & \text{if } \mathrm{c_i}= \mathrm{k}, \quad \mathrm{k} = 1,2,\ldots,\mathrm{C}, \\ -1 & \text{otherwise}, \end{cases} \tag{1.6}$$

where c_i is the class label for X_i.

In the **ELM** algorithm, the input weights (V) and bias (b) are chosen randomly for a given number of hidden neurons. By assuming the network output (Y) is equal to the coded class label (T), the output weights (W) are analytically calculated as,

$$W = YY_h^{\dagger} \tag{1.7}$$

where $Y_h^{\dagger}$ is the Moore-Penrose generalized pseudo-inverse of the hidden layer output matrix Y_h.

In summary, the simple steps involved in the **ELM** algorithm are:

- Given training samples and class labels (X_i, Y_i), select the appropriate activation function $G(.)$ and the number of hidden neurons;
- Randomly select the input weights (V), bias (b) and calculate the output weights W analytically where $W = YY_h^{\dagger}$.
- Use the calculated weights (W, V, b) for estimating the class label. We try to minimize the error between the observed and predicted values during training and select those weights which give the best classification accuracy. The final performance depends on

the choice of these parameters since overtraining or under-training can result in poor test results. These are the values that are tuned by the **PSO** algorithm.

- The estimated class label is calculated as

$$\hat{c_i} = \arg \max_{k=1,2,...,C} y_i^k. \tag{1.8}$$

Random selection of input weights (V) and bias (b) affects the performance of the **ELM** multiclass classifier significantly (Suresh et al., 2010) resulting in large variances in testing accuracies. Proper selection of **ELM** parameters (input weights, bias values, and hidden neurons) influences the performance (Saraswathi et al., 2011) of the **ELM** multiclass classifier favorably by minimizing the error defined as:

$$\{H^*, V^*, b^*\} = \arg \min_{H,V,b} \{Y - T\} \tag{1.9}$$

where Y is the observed class value and T is the calculated output value of the class, for a given set of hidden neurons H and input parameters V and b. The best weights and bias values (marked with *) for the **ELM** can be found using search techniques and optimization methods that are not very computationally intensive. In this study, we use Particle Swarm Optimization for tuning the **ELM** parameters (H, V, b).

1.4.2 Particle Swarm Optimization

Particle Swarm Optimization (Kennedy and Eberhart, 1995; Fernández-Martínez and García-Gonzalo, 2008; Fernández-Martínez et al., 2008; Fernández-Martínez and García-Gonzalo, 2010) is a global optimization algorithm that it is based on a sociological model to analyze the natural behavior of individuals in groups, such as a flock of birds that fly as a group to reach their nests. The main feature of this algorithm is its apparent simplicity when applied to solve optimization problems. The algorithm consists of the following steps:

1. Individuals, known as particles, are represented by vectors whose length is the number of degrees of freedom of the optimization problem, which is the dimension of the

problem. This is the only prior knowledge we require to solve any optimization problem. While building the model we look for solutions in this search space as shown in Figure 1.3 on page 19.

2. We start by randomly initializing the position ($\mathbf{x}_i^0$) and velocities ($\mathbf{v}_i^0$) of a population of particles. Generally, the particles try to position themselves through intelligent sampling of a prismatic volume in the model space. The velocities are the perturbations of the model parameters needed to find the global minimum (assuming that it does exist and is unique).

3. Initially the velocities are set to zero, or, they might be randomized with values not greater than a certain percentage of the search space in each direction.

4. A misfit or cost function is evaluated for each particle of the swarm in each iteration, e.g. the error between the observed and expected value could be the misfit. We might try to minimize this error and use this value to measure fitness. As time advances, the position and velocity of each particle is updated, which is a function of its own misfit and the misfit of its neighbors.

5. At time-step $k+1$, the algorithm updates positions $\left(\mathbf{x}_i^{k+1}\right)$ and velocities $\left(\mathbf{v}_i^{k+1}\right)$ of the individuals as follows:

$$\begin{aligned} \mathbf{v}_i^{k+1} &= \omega \mathbf{v}_i^k + \phi_1(\mathbf{g}^k - \mathbf{x}_i^k) + \phi_2(\mathbf{l}_i^k - \mathbf{x}_i^k), \\ \mathbf{x}_i^{k+1} &= \mathbf{x}_i^k + \mathbf{v}_i^{k+1} \end{aligned} \tag{1.10}$$

with

$$\phi_1 = r_1 a_g \quad \phi_2 = r_2 a_l \quad r_1, r_2 \to U(0,1) \quad \omega, a_l, a_g \in \mathbb{R}. \tag{1.11}$$

$\mathbf{l}_i^k$ is the best position found so far by i^{th} particle and $\mathbf{g}^k$ is the global best position with respect to the whole swarm (or within a neighborhood if a local topology is used). ω, a_l, a_g are called the inertia and the local and global acceleration constants, and these are the parameters we have to tune for the **PSO** to achieve convergence. $r_1,\ r_2$ are uniform random numbers used to generate the stochastic global and local accelerations,

ϕ_1 and ϕ_2. Due to the stochastic effect introduced by these numbers **PSO** trajectories should be considered as stochastic processes. The deterministic trajectories of the **PSO** are fully analyzed in (Fernández-Martínez et al., 2008), which is important to understand the capabilities of the **PSO** algorithm.

Particle Swarm Optimization is a double discrete gradient method as shown in Figure 1.3 on page 19, with random effects introduced in the global and local acceleration constants, by uniform random numbers r_1, r_2. The physical representation of the **PSO** algorithm can be interpreted as a damped mass-spring system as shown in Figure 1.4 on page 20, with unit mass, damping factor, $1-\omega$, and stochastic stiffness constant, ϕ (Fernández-Martínez and García-Gonzalo, 2008):

$$\begin{cases} \mathbf{x}_i''(t) + (1-\omega)\,\mathbf{x}_i'(t) + \phi \mathbf{x}_i(t) = \phi_1 \mathbf{g}\,(t) + \phi_2 \mathbf{l}_i\,(t)\,,\ t \in \mathbb{R}, \\ \mathbf{x}_i(\mathbf{0}) = \mathbf{x}_{i0}, \\ \mathbf{x}_i'(\mathbf{0}) = \mathbf{v}_{i0}, \end{cases} \tag{1.12}$$

In this model the force term is composed of the global and local attractors, $\mathbf{g}\,(t)$ and $\mathbf{l}_i\,(t)$. The spring-mass analogy was used (Fernández-Martínez and García-Gonzalo, 2008) to derive the generalization of Particle Swarm Optimization (**GPSO**), for any iteration time and discretization step, as:

$$\begin{aligned} &\mathbf{v}_i(t+\Delta t) = (1-(1-\omega)\,\Delta t)\,\mathbf{v}_i\,(t) + \phi_1 \Delta t\,(\mathbf{g}\,(t) - \mathbf{x}_i(t)) + \phi_2 \Delta t\,(\mathbf{l}_i\,(t) - \mathbf{x}_i(t))\,, \\ &\mathbf{x}_i(t+\Delta t) = \mathbf{x}_i(t) + \Delta t \mathbf{v}_i(t+\Delta t),\ t,\ \Delta t \in \mathbb{R} \\ &\mathbf{x}_i\,(0) = \mathbf{x}_{i0},\ \mathbf{v}_i\,(0) = \mathbf{v}_{i0}. \end{aligned} \tag{1.13}$$

Using different finite differences schemes to approach $\mathbf{x}_i''(t),\ \mathbf{x}_i'(t)$ the authors (Fernández-Martínez and García-Gonzalo, 2009, 2010) have introduced different **PSO** versions of the same family known under the following acronyms: **CC-PSO** for Centered-Centered PSO, **CP-PSO** for Centered-Progressive PSO, **RR-PSO** for Regressive-Regressive PSO and **PP-PSO** for Progressive-Progressive PSO. These families use a concept called **cloud-PSO** which enables the particles to have different explorative and exploitative capabilities for better performance. These versions of **PSO** families were tested out in this study and were found to be robust in

all applications. In all the cases the algorithm convergence is related to the stability of the first and second order moments of the particle trajectories considered as stochastic processes.

The performance of each algorithm will depend on the degree of numerical difficulties of the cost function and the number of dimensions. In general, **RR-PSO, CC-PSO** and **GPSO** are the most exploitative (search deeply in a small area) versions while **CP-PSO** and **PP-PSO** are the most explorative (search far and wide in a bigger area). In particular, **RR-PSO** has very good exploration capabilities, due to the way the algorithm updates its position and velocity. In **PSO** which is ordinarily used, the velocity is updated first and then the position is updated. **CC-PSO** updates the position first and then the velocity using two consecutive attractors' positions. **RR-PSO, CP-PSO** and **PP-PSO** update positions and velocities at the same time and are the more explorative versions. When numerical difficulties increase, exploration might be needed and the **CP-PSO** could eventually provide very good results, which might be even better than the more exploitative versions which can be get trapped either in a local minima or in a flat area.

For different benchmark functions the parameter sets with a high probability of success are close to the upper limit of second order stability, where the exploration is very high since the variance of the trajectories becomes unbounded. Based on this idea the authors (Fernández-Martínez and García-Gonzalo, 2008) have designed the *cloud-PSO* algorithm, where each particle in the swarm has different inertia (damping) and acceleration (rigidity) constants (García-Gonzalo, 2009). This work has been recently expanded to other versions of the **PSO**-family (Fernández-Martínez and García-Gonzalo, 2010). This feature allows the **PSO** algorithm to control the velocity update and to find sets of parameters that are better suited for individual optimization problems, where some of the particles will be more explorative while others will have higher exploitative character. In this algorithm design of the Δt parameter arises as a natural numerical constriction factor to achieve stability. When this parameter is less than one, the exploration around the global best solution is increased. Conversely when Δt is greater than one, the exploration of the whole search space is increased, helping to avoid entrapment in local minima. This feature was used by the au-

thors to create the lime and sand algorithm that combines different values of Δt depending on the iterations used (Fernández-Martínez and García-Gonzalo, 2008). All these methods were applied during this thesis study at different times on various data sets. We saw typically, an improvement of 5 to 8 % in classification accuracy after applying these modified and improved **PSO** methods, compared to previous performances with the traditional **PSO**.

1.5 Thesis organization

In this thesis, we present our two publications on protein secondary structure prediction (Saraswathi et al., 2010b) and prediction of protein relative solvent accessibility (Saraswathi et al., 2010a). The algorithm for the data development was done by Pawel Gniewek, as acknowledged in the papers. Saraswathi, S. generated the data and developed the machine learning algorithms and optimization methods to obtain the final results. Follow up studies were conducted by Saraswathi.S, on a variety of protein data and the improved optimization techniques have resulted in higher accuracies for protein secondary structure and function predictions. The results of these studies, which are almost ready for submission to journals, are also presented here. In essence, this thesis has five distinct parts.

1. Part I: Chapter 1: GENERAL INTRODUCTION: A discussion of the various studies presented in this thesis, specific aims and thesis organization.
2. Part II: PROTEIN SECONDARY STRUCTURE PREDICTION:
   (a) Chapter 2 is an introduction to secondary structure prediction.
   (b) Chapter 3 is an initial study (published) for secondary structure prediction using knowledge-based potentials data.
   (c) Chapter 4 presents the same study with much improved results using the same data but an advanced and optimized **FLOPRED** algorithm.
   (d) Chapter 5 presents an amino acid perspective of the results obtained using FLOPRED algorithm.

(e) Chapter 6 presents a study of secondary structure prediction using a new database of physicochemical properties of amino acids.

(f) Chapter 7 presents possibilities to improve the results of secondary structure prediction using position specific residue preferences of amino acids at the ends of secondary structures.

(g) Chapter 8 draws some general conclusions on secondary structure predictions.

3. Part III: Chapter 9: PROTEIN RELATIVE SOLVENT ACCESSIBILITY PREDICTIONS: Results and analysis of **RSA** predictions (published) using protein sequences as input data are given in Chapter 9.2 on page 161.

4. Part IV: Chapter 10: PROTEIN PHOSPHORYLATION PREDICTIONS: Results and analysis using a new set of data for phosphorylation prediction are given in Chapter 10.3 on page 173.

5. Part V: Chapter 11: GENERAL CONCLUSIONS: Discussion of overall results and future plans are given in Chapter 11 on page 176.

1.6 Collaborative work

Data generation

The *data generation algorithm for the knowledge-based potentials data*, used in chapters 3 and 4 and the *data for the initial study of* 84 *proteins* was generated by Pawel Gniewek, a summer student in our lab. Saraswathi, S. collaborated with Pawel Gniewek in testing the results of the algorithm during development. Pawel Gniewek was under the supervision of Prof. Robert Jernigan (Iowa State University), his professor Dr. Andrzej Koliniski (Faculty of Chemistry, Warsaw University, Warsaw) and Prof. Andrzej Kloczkowski (Ohio State University, USA). All other data generated for other studies were developed by Saraswathi, S.

Toolboxes

Simple **PSO** and **GA** toolboxes (Suresh et al., 2010; Saraswathi et al., 2011) used in Chapters 3, and 9 respectively, and **S-ELM** (Suresh et al., 2010; Saraswathi et al., 2011) in chapter 9, is a modified version of the original open source **ELM** (Huang et al., 2006). These software were developed by Prof. Suresh Sundaram of Nanyang Technological University, Singapore and Saraswathi, S. was involved in testing of these toolboxes during development. These were used in an earlier collaborative work (Saraswathi et al., 2011; Suresh et al., 2010).

ELM toolboxes used in all other chapters such as **(FLOPRED)** are modified versions of the original open source **ELM** tool box (Huang et al., 2006; Saraswathi et al., 2010b) and **S-ELM** (Saraswathi et al., 2011; Suresh et al., 2010). In **FLOPRED**, the hidden neuron parameters for **ELM** are tuned by the advanced **PSO** algorithms instead of being user defined as in previous **ELM** versions.

Advanced **PSO** and **PCA** toolboxes used in Chapter 4 through 9 (except for **PSO** in Chapter 8), were initially developed by Prof. Luis Fernández-Martínez (Fernández-Martínez and García-Gonzalo, 2010, 2009; Fernández-Martínez et al., 2008). These **PSO** algorithms were improved in this study in collaboration with Saraswathi, S. to include **ELM** and **PSO** parameters in the tuning of advanced PSO algorithms. The publicly available **WEKA** toolbox (Witten and Frank, 2005) was used for the **SVM** studies.

Research and Analysis

All advanced **PSO** algorithm studies and the **PCA** studies were conducted in collaboration with Prof. Juan Luis Fernández-Martínez (University of Oviedo, Spain). All other data generation, development of machine learning algorithms and analysis of results were performed only by the author of this thesis (Saraswathi), under the guidance of her professors Dr. Robert Jernigan and Prof. Andrzej Kloczkowski.

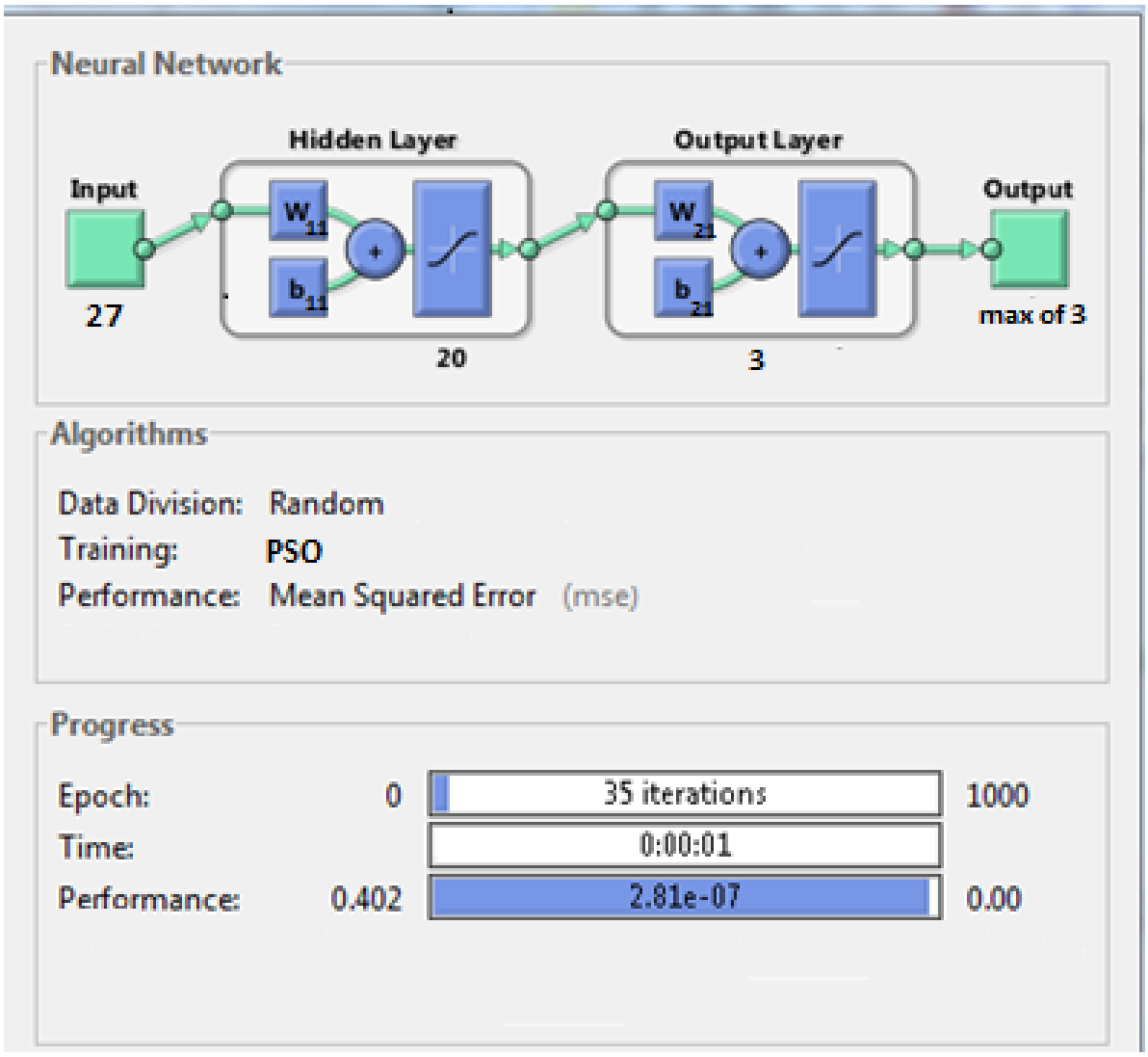


Figure 1.2 A traditional neural network

This figure shows a Neural Network (**NN**) classification in progression, that has 27 features as inputs, 30 hidden neurons (only one shown) and three units of output (only one shown) for the three secondary structure classes. Data can be selected at random as residues from a single or multiple proteins or all residues of a single protein can be selected for training or testing purposes. In a Traditional **NN**, conjugate gradient method is used for optimization but in **FLOPRED** we use **PSO** instead. The number of iterations is set to 1000 in this figure and 35 of them have been completed as shown here. The gradient or **PSO** value (minimum error) is evaluated during each iteration. The performance gives the $\mathbf{Q}_3$ accuracy obtained (so far) for the classifications. The time used up until this point for these calculations are also shown.

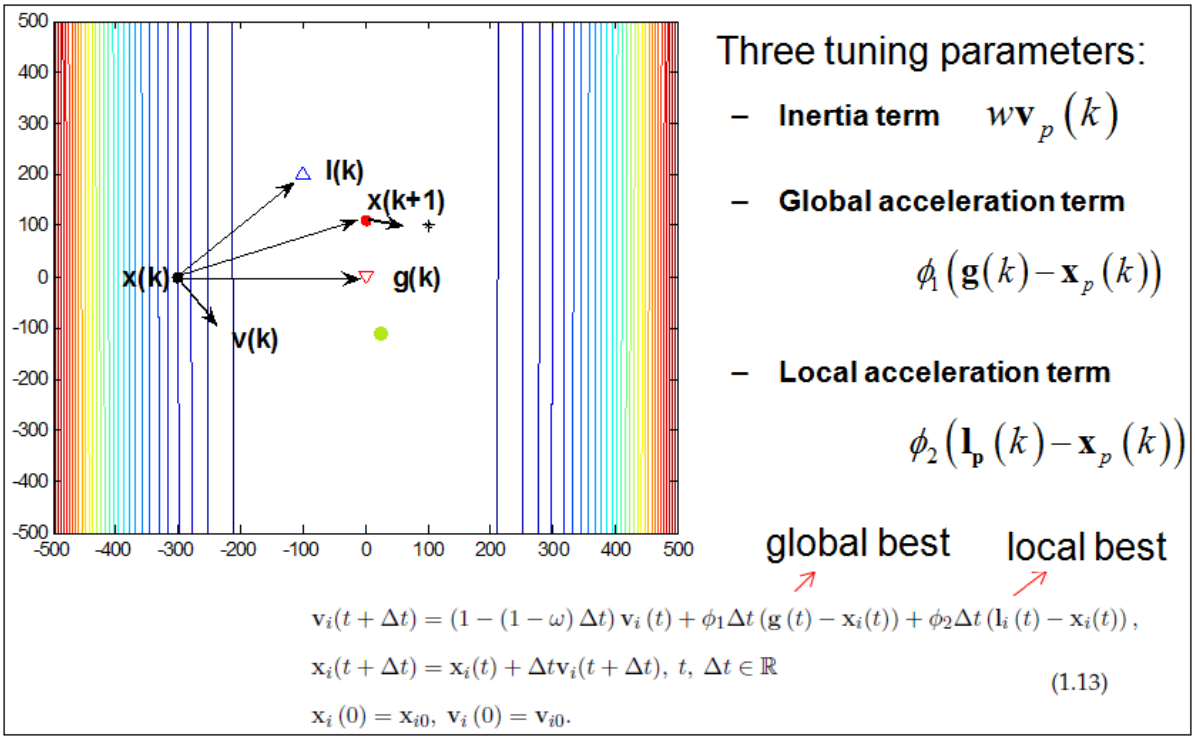
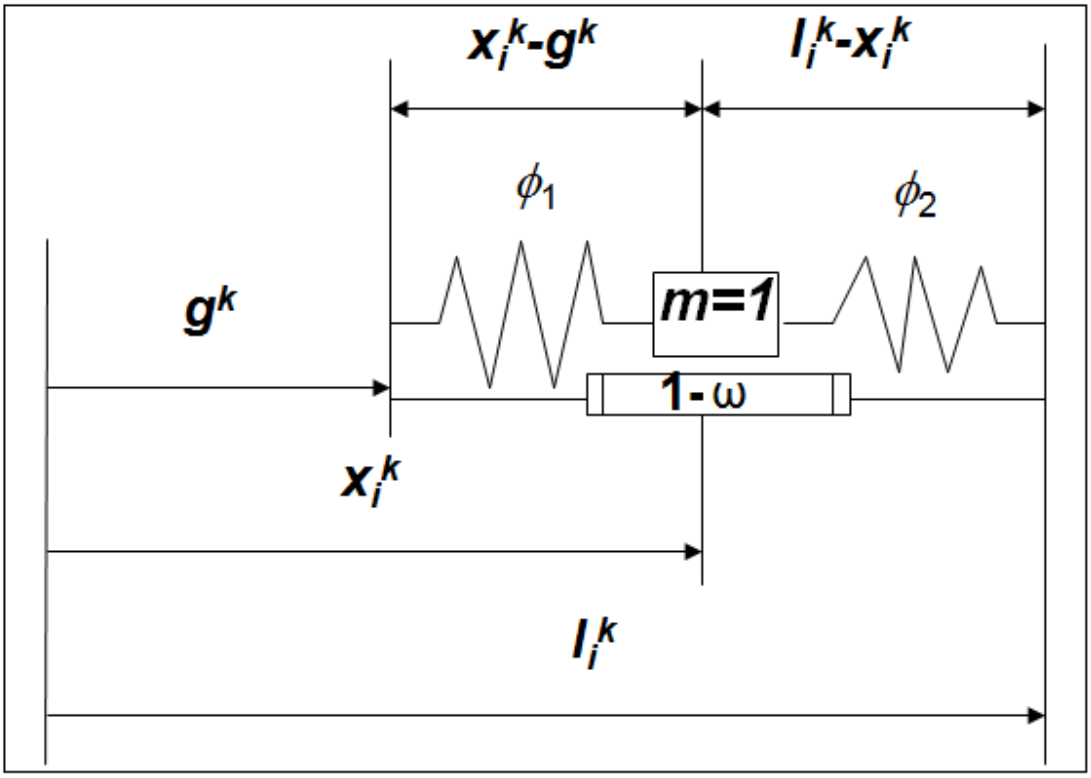


Figure 1.3 **PSO**: a stochastic two-discrete gradient

The particle swarm optimization algorithm has three tuning parameters; the inertia term, the global acceleration and local acceleration term which are iteratively updated according to the formula shown. They are updated with respect to their local and global best positions and the stochastic acceleration terms ϕ_1 and ϕ_2 and learning rate ω. The green dot denotes the optimum desired value, the red dot is the next updated position for the particle X_i in the k^{th} iteration, while all the blue dots represent the swarm particles which are trying to achieve the global minimum by getting closer to the green dot, which is the ultimate desired position. The higher red streaked regions are far away from the global minimum while the lighter color streaked lines show areas closer and closer to the desired global minimum. The particles converge to the global minimum after a set number of iterations. (*Slide printed with permission from Prof. Juan Luis Fernández-Martínez.*)



$$x_i''(t) + (1-\omega)\cdot x_i'(t) + (\phi_1 + \phi_2)\cdot x_i(t) = \phi_1 \cdot l_i(t - t_0) + \phi_2 \cdot g(t - t_0)$$

Figure 1.4 **PSO**: Spring - Mass analogy

Particle Swarm Optimization algorithm is represented here as a damped spring system. The trajectory of each particle mimics the motion of a unit mass, **m**, attached to two springs with rigidity constants ϕ_1 and ϕ_2, and damping (1-ω), whose equilibrium positions are $l_i(t)$, which gives the individual best position for each particle and $g(t)$,which gives the global best position in the swarm. The given equation represents the damped spring system as a difference equation. (*Slide printed with permission from Prof. Juan Luis Fernández-Martínez.*)

REFERENCES

Duan, M., Huang, M., Ma, C., Li, L., and Zhou, Y. (2008). Position-specific residue preference features around the ends of helices and strands and a novel strategy for the prediction of secondary structures. *Protein Science*, 17:1505–1512.

Fernández-Martínez, J. L. and García-Gonzalo, E. (2008). The Generalized PSO: A New Door to PSO Evolution. *Journal of Artificial Evolution and Applications*, 2008:15.

Fernández-Martínez, J. L. and García-Gonzalo, E. (2009). The PSO family: deduction, stochastic analysis and comparison. *Special issue on PSO. Swarm Intelligence*, 3:245–273.

Fernández-Martínez, J. L. and García-Gonzalo, E. (2010). Two algorithms of the extended PSO family. In *Proceedings of IJCCI/ICNC*, pages 237–242.

Fernández-Martínez, J. L., García-Gonzalo, E., and Fernández-Alvarez, J. P. (2008). Theoretical analysis of particle swarm trajectories through a mechanical analogy. *International Journal of Computational Intelligence Research*, 4:93–104.

Fernández-Martínez, J. L., Mukerji, T., and García-Gonzalo, E. (2010). Particle swarm optimization in high dimensional spaces. *Swarm Intelligence*, 77:496–503.

García-Gonzalo, E. a. J. L. (2009). Design of a simple and powerful particle swarm optimizer. In *Proceedings of the International Conference on Computational and Mathematical Methods in Science and Engineering*, pages 1280–1290.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, New York.

Huang, G. B. and Babri, H. A. (1998). Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions. *IEEE Trans. Neural Networks*, 9:224–229.

Huang, G. B., Zhu, Q. Y., and K, S. C. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, 70:489–501.

Kawashima, S., Ogata, H., and Kanehisa, M. (1999). AAindex: Amino Acid Index Database. *Nucleic Acids Research*, 27:368–369.

Kennedy, J. and Eberhart, R. C. (1995). Particle Swarm Optimization. *Proceedings of the IEEE International Conference on Neural Networks*, 4:1942–1948.

Pearson, K. J. (1901). Principal Components Analysis. *The London, Edinburgh and Dublin Philosophical Magazine and Journal*, 6:566.

Richardson, C. J. and Barlow, D. J. (1999). The bottom line for prediction of residue solvent accessibility. *Protein Engineering Design and Selection*, 12:1051–1054.

Saraswathi, S., Jernigan, R. L., and Kloczkowski, A. (2010a). An Extreme Learning Machine Classifier for prediction of relative solvent accessibility in proteins. *Proceedings of IJCCI/ICNC*, pages 364–369.

Saraswathi, S., Jernigan, R. L., Koliniski, A., and Kloczkowski, A. (2010b). Protein secondary structure prediction using knowledge-based potentials. *Proceedings of IJCCI/ICNC*, pages 370–375.

Saraswathi, S., Suresh, S., and Sundararajan, N. (2011). Icga-pso-elm approach for accurate multiclass cancer classification resulting in reduced gene sets in which genes encoding secreted proteins are highly represented. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 8:452–463.

Suresh, S., Saraswathi, S., and Sundararajan, N. (2010). Performance enhancement of extreme

learning machine for multi-category sparse cancer classification. *Engineering Applications of Artificial Intelligence*, 23:1149–1157.

Witten, I. H. and Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco.

PART II

PROTEIN SECONDARY STRUCTURE PREDICTION

CHAPTER 2. INTRODUCTION TO SECONDARY STRUCTURE PREDICTION

2.1 Background and significance

2.1.1 Motivation

It is important to have a deep understanding of protein functions in order to maintain and improve the quality of life for all living organisms. A protein's three-dimensional structure (3-D) determines its function. Prediction of secondary structure is a useful intermediate step to speed up the process of determining or predicting 3-D structure (Lomize et al., 1999; Ortiz et al., 1999), since proteins form local conformational patterns like residual α-helices and β-strands that eventually fold up into the 3-D structure. Knowledge of secondary structures of proteins can help in the identification and classification of protein 3-D structures (Liu and Wang, 2007) and functional motifs, help with structure alignments in homology modeling (Krissinel and Henrick, 2004; Wray and Fisher, 2007), help in investigating gene functions and in sequence annotation. It has been difficult to predict structures for non-homologous sequences or for those sequences with weak homology to known structures (Rost, 2001; Zhang et al., 2011). Existing 3-D structure prediction methods like homology modeling, have been successful in determining the structures for newly discovered protein sequences when there is greater than 30% sequence identity (Rost, 2001). But only about 40% of all available sequences can be annotated using this method. Even in these cases, structures for only parts of sequences can be determined. For higher Eukaryotic organisms like Eukaryotes which have long protein sequences it is even more difficult to obtain whole structures or even parts of the structures in comparison with prokaryotes. Since structures are thought to be more conserved than sequences, it is possible that non-homologous sequences might share the same structure.

So information on secondary structures is even more important when we need to determine structures of sequences where there is low homology (0% to 30%). Secondary prediction can also help to determine structures for membrane proteins where very few 3-D structures for those proteins are currently known (Kashlan et al., 2006). Computational methods will be invaluable because prediction methods can help find structures and functions of proteins where other methods fail. Further improvements in secondary structure prediction can lead to progress in protein engineering, drug design and many other areas of applications. Hence prediction of secondary structures and other related protein features from protein sequences is an important subject among researchers.

Advances in mass-scale genome sequencing technologies have resulted in the availability of millions of protein sequences. There are almost $11,934,213$ protein sequences belonging to $11,536$ organisms that are available to date according to RefSeq Release 45 (Pruitt et al., 2009) while we have only 71635 known protein structures (Tuesday Mar 08, 2011 at 4 PM PST), with an average yearly growth of just 7000 structures, according to **RSCB PDB** (Berman et al., 2000). There is a big gap that needs to be filled in terms of protein structure determination. Protein structures determined through experimental techniques such as X-ray crystallography and Nuclear Magnetic Resonance **(NMR)** are expensive and time consuming for processing on the genome scale. Computational methods can predict secondary structure in a much shorter time frame. Machine learning methods are useful for this purpose and once the training models are built from existing information, which might take at most a few months, structure prediction can be done at much lower cost. Protein secondary structure prediction has gained increasing importance in computational biology due to this growing demand for large scale structure prediction. Hence there is a need for faster and cheaper computational methods (such as machine learning) that can predict a protein's structure much more efficiently and less expensively, with acceptable levels of accuracy.

Details necessary for studies conducted in more than one chapter, under secondary structure prediction **(SSP)**, are given in this introductory chapter. The subjects to follow are:

- Protein structures.

- Factors that determine protein secondary structure.
- Data encoding techniques.
- Secondary structure assignments **SSP**.
- Computational methods used in the literature for **SSP**.
- Secondary structure accuracy.
- Limits of secondary structure predictability.
- Knowledge recovery from secondary structure predictions.
- Description of generation of knowledge-based potentials with **CABS** (C-α-C-β-Side group protein model described on page 46).
- References.

2.1.2 Protein structures

All proteins begin life as a string of amino acids, which constitute their primary structure. Protein sequences are made of amino acid residues that have different physical and chemical properties such as charges, polarities, heterogeneities, and many other such features, which enable them to form different secondary structures such as α-helices (**H**), β-strands (**E**) and coils (**C**).

The chemical structure of an amino acid consists of an amine group ($\mathbf{NH_2}$), an α-carbon and a carboxylic acid (**COOH**), which are the common units in all amino acids. In addition a side-chain '**R**' is attached to its α-carbon, which differentiates each of the 20 amino acids. The side-chains have different physicochemical properties such as size, polarity, charge and hydrophobicity. These properties influence an amino acids' interaction with other amino acids and the solvent environment. These interactions in turn influence the proteins' folding and functions. Thus, the study of amino acid composition and arrangement in a protein sequence is important.

Protein structures are of four types:

- *Primary structures*: are polymers of 20 different types of amino acids. The amino acids in a protein sequence are covalently bonded by peptide bonds between two adjacent residues. The common units of all the amino acids in the polymer form the backbone of the protein while the side-chains protrude away from the protein chain, as seen in Figure 2.2, which shows a few residues (30 to 35) from the ubiquitin protein. This string of amino acids is known as the primary sequence of a protein, which extends from the N-terminal or amino end to the C-terminal or carboxyl end. In some places disulphide bonds help to stabilize the proteins by bridging between cysteines.

- *Secondary structures* are formed from the primary protein sequences through local hydrogen bonding between the backbones, bringing stability to the protein structure. Some examples of secondary structures are α-helices and β-strands, as seen in Figure 2.3.

- *Tertiary structures* are the 3-dimensional folding of a single polypeptide chain due to interaction among the residues. Disulphide bonds between cysteine residues may also be present. Super-secondary structures, which are almost like tertiary structures, are distinguished by peculiar arrangements of two or three secondary structures and are found in different types of protein structures with completely different protein sequences. For example, Coiled Coils, EF hand and Tim Barrels as seen in Figure 2.4 are such stuctures. These motifs play a very important roles in drug design since infectious elements such as HIV viruses use these structures to enter the cells of their victims.

- *Quaternary structures* have two or more polypeptide chains with their own tertiary structures that form a multi-subunit structure, which are stabilized by non-covalent interactions, as seen in Figure 2.5 and 2.6.

2.1.3 Factors that determine protein structure

The order and variety of the primary sequences of amino acids is believed to play a major role in determining a protein's 3-D folded structure and function. Secondary structures and their solvent environment determine the final tertiary or quaternary structures into which the

proteins will eventually fold into. In the ideal case, it is believed the protein sequences and the intermediary secondary structures play a pivotal role in determining the final structure and function of a protein. Other viewpoints suggest that some folding may proceed sequentially as synthesis on the ribosome occurs or that some folds may occur because of kinetic trapping. These alternative mechanisms may account for some errors in secondary structure prediction. Biotechnological advances provide exponentially increasing volumes of precise sequence information that can be used by computational techniques to predict protein 2-D and 3-D structures. There is a vast amount of information that can be obtained from the protein data bank (Berman et al., 2000) and other databases such as :

- Secondary structures **(DSSP, STRIDE)** , Relative solvent accessibility (**RSA**) of proteins,
- Multiple Sequence Alignments **(MSA)**, post-translational modification **(PTM)** regions,
- Disordered regions **(DR)** or dual-personality **(DP)** regions of proteins and
- Position-specific residue preference **(PSRP)** of amino acids at the ends of secondary structures

These data have contributed to much improved computational techniques and better assessments of accuracies in secondary structure predictions. With increasing sequence information, secondary structure prediction has renewed the interests of many researchers and is the focus of many of the studies in this thesis.

2.1.4 Data encoding for secondary structure prediction

Windows of protein sequences encoding local amino acid interactions have primarily been used for secondary structure prediction. Protein sequences are coded using an orthogonal binary representation of the twenty amino acids as a 20-element binary vector, where the residue of interest is coded as 1 and all other amino acids are coded as $0s$. Specification of a window size determines the number of amino acids considered to influence the central residue of interest in its local interaction with neighboring residues that may influence

secondary structure formation. In lieu of binary coding, sequences are sometimes encoded using position specific scoring matrices (**PSSM**) (Jones, 1999), which also encodes evolutionary information from **MSA**. These **MSA** have been popularly used for predicting secondary structures, which resulted in improved prediction accuracies (Zvelebil et al., 1987). The ever-growing databases of known protein sequences, on which more recent studies are based, have also helped to improve prediction accuracies. With the increasing availability of newer and more diverse protein structures, current studies increasingly try to use protein structure information, in addition to newer sequence and evolutionary information to take advantage of these vast resources, to further improve secondary structure prediction. These limited successes have encouraged us to use structural information encoded in the features of protein sequences. This new data set does not use the traditional orthogonal coding, as in **PSSM** or **MSA**. This new data captures the knowledge-based potential information embedded in the amino acid sequences, calculated using the **CABS** algorithm (Kolinski, 2004), which captures structural information by predicting probable structures. We have made sure that there is no structural similarity between the templates used for building this structural data set and the actual sequences used for modeling and testing our results.

2.1.5 Secondary Structure Assignment

The Database of Secondary Structure in Proteins (**DSSP**), (Kabsch and Sander, 1983), provides consistent secondary structure assignments to all known proteins, and these assignments are widely accepted. These differ from the assignments given in the **PDB** database by experimental crystallographers or **NMR** scientists. According to the **DSSP** classification, there are eight types of secondary structure which can be assigned: **H** (α-helix), **E** (extended β-strand), **G** (3_{10} helix), **I** (π-helix), **B** (bridge, a single residue β-strand), **T** (β-turn), **S** (bend), and **C** (coil). These assignments are calculated from the hydrogen bonds that form between the backbone carbonyl **(CO)** and amino **(NH)** groups. Several groups have interpreted these assignments in different ways (Kloczkowski et al., 2002) depending on the grouping of amino acids into one of several smaller groups. It was shown (Rost, 2001) that secondary structure

prediction accuracies can show higher accuracies compared to other interpretations if 3_{10} helices and β-bulges structures are interpreted as coil (C). Alternate assignments of secondary structure assignments include **STRIDE** (Frishman and Argos, 1995) and **KAKSI** (Martin et al., 2005). Commonly, secondary structure prediction is based on three structure types: α-helix **(H)**, β-sheets **(E)** and Coil **(C)**. In our study, we use a standard 3-class secondary structure assignment where Helix (H) in the three letter code includes the three **DSSP** states **H, G,** and **I**; β-strand **(E)** contains **E** and **B**; and coil**(C)** consists of **T, S, Blanks** and **C**.

2.1.6 Computational methods for protein secondary structure determination

Secondary structures can provide complementary information which might be difficult to obtain by experimental means (Oklejas et al., 2010). Study of secondary structures through various computational means has been a popular topic for research. Many researchers have been trying to improve prediction methods and accuracies. A discussion on the evolution of computational methods for secondary structure prediction follows next.

2.1.6.1 Historical view of secondary structure prediction

Several computational methods have been used successfully for secondary structure prediction. The most common methods used for secondary structure prediction are

- Empirical statistical methods
- Hidden Markov models
- Nearest neighbor methods
- Neural network methods

Chou and Fasman were the pioneers in the field of secondary structure prediction and used empirical methods based on the simple relative frequencies of amino acids in secondary structure for prediction of protein secondary structures (Chou and Fasman, 1974). The popular **GOR** prediction methods for secondary structure prediction, (Chou and Fasman, 1974;

Garnier et al., 1978, 1996; Zvelebil et al., 1987) were based on information theory and Bayesian statistics. This method was further improved by the same group through several subsequent **GOR** versions. Evolutionary information was used in **GOR V** (Kloczkowski et al., 2002) for improved structure prediction using multiple sequence alignments. Nearest neighbor algorithms were used by several groups (Salzberg and Cost, 1992; Yi and Lander, 1993; Salamov and Solovyev, 1995; Salamov, 1997). Support Vector Machines (**SVM**), based on statistical Learning Theory (Vapnik, 2000), were also used (Ward et al., 2003) for secondary structure prediction by many researchers.

Machine learning methods, particularly neural networks which are used in this study, have proved to be most successful among all methods used for secondary structure prediction. Hence, our discussion and comparison of results mostly relate to neural network studies in literature. Early Neural Network based secondary structure predictors (Qian and Sejnowski, 1988) were followed by numerous other studies as discussed below. Further improvements in structure prediction came when multiple sequence alignments **(MSA)** were introduced by several groups (Rost and Sander, 1993; Rost, 1996; Cuff and Barton, 2000; Kloczkowski et al., 2002), yielding around 70% accuracy. The most successful prediction algorithms commonly used today are the **PHD** method (Rost, 1996) and **PSIPRED** (Jones, 1999) yielding over 76% accuracy. The highly successful prediction algorithm, PredictProtein server (Rost et al., 2004) uses **MSA** based neural networks. The **PSIPRED** algorithm (Jones, 1999) relies on **PSI-BLAST** Altschul et al. (1997) and neural networks to obtain better than 80% prediction accuracy. Increasing availability of protein sequences has also helped to build better models and attain higher accuracies. The **Jpred** (Cuff and Barton, 2000) prediction server runs on the **JNet** algorithm and can predict three types of secondary structures (α-helix, β-strand and coil) for an accuracy of 81.5%, using **PSI-BLAST** (Altschul et al., 1997), a Position-specific scoring matrix **(PSSM)** and **HMMER** (Eddy, 1998) using Hidden Markov Model **(HMM)** profiles. Despite the many different methods and complicated algorithms used for secondary structure predictions, the accuracies measured by $\mathbf{Q_3}$, have hardly exceeded 70% range for methods that use stand alone algorithms and single sequences. The threshold of 88% has

usually been surpassed for methods that include **MSA** as part of the prediction algorithm. $\mathbf{Q_3}$ gives the average accuracy for all three commonly used secondary structure classes as explained in Section 2.1.7 on page 35. The difficulty in attaining higher accuracies might possibly be due to the exclusion of long-range interactions in the data that is used (Kihara, 2005). More recently long range interactions were implemented (Madera et al., 2010) on a k-mer order model and Markov chains were used to obtain prediction results of 77.4% with a standard deviation of 0.2%. They also claimed to have improved the quality of prediction with an increased Segment OVerlap **(SOV)** score which was 1.8% more than previously reported results of 80.5%. We look at more recent studies next.

2.1.6.2 Recent studies in secondary structure prediction

Encouraged by the availability of biological information and better computing resources, larger and more complicated algorithms are being built to achieve better prediction accuracies, but still have not been able to go much beyond the virtual accuracy barriers discussed earlier. Recent studies were interested in long-range interactions of amino acids (Kihara, 2005) and their effects on secondary structure formation. These studies suggested that long-range interactions can potentially play an important role in achieving higher classification accuracy. Secondary structure prediction methods using a large number of resources, were developed (Montgomerie et al., 2006; Pollastri et al., 2007) using structural frequency profiles from existing PDB templates and high-throughput machine learning systems and reported an accuracy of 85.7%, one of the highest accuracies reported so far. Three independent expert neural networks were used (Sivan et al., 2007) for the three secondary structures at the first level. Chou and Fasman frequency values were used at the second level. This method helped reduce the search space for experimental methods. The results of classifications from several secondary structure prediction servers were studied and analyzed (Kazemian et al., 2007) to discern patterns of prediction accuracies with respect to different amino acids. The authors gave an analysis of these results from an amino acid perspective based on the results of the servers. We have likewise done an in-depth analysis of the amino acids accuracies obtained from our

own results and saw some interesting and intriguing patterns in the classification results. We share these results in section 5.1.1 page 101 of this thesis.

Ghosh and Parai , (Ghosh and Parai, 2008) used three distance-based classifiers on protein sequences which are coded as features to represent patterns of neighboring residues instead of traditional binary numbers. They found that minimum distance classifiers perform better than **K-NN** and fuzzy **K-NN** classifiers, with a best accuracy of 59.21%. They drew attention to the fact that there needs to be a better representation for amino acid sequences, which takes into account the proximity of amino acid residues to the central residue of interest. A two-stage algorithm was used (Yuksektepea et al., 2008) to predict secondary structures from protein sequences, where they used a mixed-integer linear programming **(MILP)** to determine the fold-type of the target sequence in the first stage. At the second stage, a probabilistic approach was used to determine secondary structure classes of the target sequences, to get an overall accuracy of 74.1%. An **ELM** approach is used (Wang et al., 2008) for secondary structure binary classification. Then a Probability Based Combination **(PBC)** method and helix post-processing was used to combine these predictions for secondary structure classification to yield an accuracy of 71.2% on the **CB513** set of protein sequences. Another knowledge-based secondary structure prediction method (Yang et al., 2009) called **KAAPRO** (**KDD*** Association Analysis PROtein secondary structure prediction), used a multi-hierarchical pyramid prediction model which classifies α-helix and β-strand proteins at an accuracy of 74.6% for a very small set of 4 proteins. A combination of three neural networks was used (Malekpour et al., 2009) on sequences encoded as multiple sequence profiles and used Segmental Semi-Markov Models **(SSMMs)** as the decision function to discriminate between secondary structures, with an accuracy of 75.18%. Palopoli et al., used an ensemble of predictors by combining the prediction results of several prediction servers (Palopoli et al., 2009), tailored to particular aspects of the target protein. They showed improvements in the prediction results of several individual proteins as compared to previous results.

Computational methods such as **GOR**, **DPM** and **Predator** were used by the **ANTHEPROT** server (Santiago-Gómez et al., 2010) to compare experimental results of secondary

structure predictions. Zhou and Yang et al. used a Knowledge Discovery **(KDD)** approach and proposed a Structural Association Classification **(SAC)** approach to secondary structure classification (Zhou et al., 2010), which used high and low confidence information to form over 8000 rules divided into three rule sets, one each for the three basic secondary structures. Through these rules they developed a Classification based on Multiple Association Rules **(CMAR)** algorithm for secondary structure prediction yielding an accuracy of 80.49% for the **CB**513 set. An attempt to capture distant interactions between amino acids residues (Bidargaddi et al., 2009) has been carried out using generative models based on Bayesian segmentation and generalized Hidden Markov Models with explicit state duration. They used a neural network and optimization methods in the second stage for secondary structure classification using *only* protein sequence data to obtain an accuracy of 71%. Modular Reciprocal Recurrent Neural Networks **(MRR-NN)** were used (Babaei et al., 2010) to model short-range interactions and a Multilayer Bidirectional Recurrent Neural Network **(MBR-NN)** was introduced to capture the long-range intramolecular interactions between amino acids. These two networks were used to capture the secondary structure patterns of amino acids with an accuracy of 79.36%. A two-level Mixed-Modal Support Vector Machine (**MMS**) was used (Yang et al., 2011) for secondary structure prediction by using physicochemical properties of amino acids and position-specific scoring matrices (PSSM) generated from **PSI-BLAST** (Altschul et al., 1997). Use of PSI-BLAST helps to include evolutionarily divergent information and conserved residue information on a longer range, contributing to increased accuracies. They integrated the **MMS** module with a modified Knowledge Discovery in Databases **(KDD*)** process and a Mixed-Modal Back Propagation neural network **(MMBP)** module to achieve accuracies of up to 85.6%, one of the highest accuracy reported so far.

2.1.7 Secondary Structure Accuracy Measures

Secondary structure prediction results classify amino acids as belonging to one of three secondary structure classes, α-helix, β-sheet or coil. There are certain performance measures that are traditionally used for the per-residue accuracy level, such as Sensitivity, Specificity

and Matthew's correlation coefficient. Although these measures are commonly used, we want to take a closer look at some issues regarding the reliability of these measures (Altman and Bland, 1994). We would like to highlight these matters to point out that these values can vary widely if the underlying composition of residues in the dataset is not well represented in the training models. We discuss some aspects of these issues and suggest ways in which we can gain a better understanding of the reliability of these measures. This understanding will allows us to build better models yielding better classifications. Some other measures such as **SOV** and J^{scores} which pertain to reliability of predictions with respect to segments of secondary structures are also discussed. These metrics are illustrated in Table D.1. All of these discussions include the assumption that α-helix is the positive class and β-sheet and coil are the negative classes, where the results are combined for these latter two classes. Similar arguments can be made by considering the other two classes as the positive class. All quantities that are used to calculate the accuracy measures given below are defined in Appendix D on page 186.

2.1.7.1 Post-test odds

$$\mathbf{odds_{post}} = odds_{post} * LRN \tag{2.1}$$

Odds$_{post}$ incorporates four different kinds of information such as prevalence, nature of the training samples, pre-test odds and the results of the test itself to determine the chances that the classification results actually belong to the positive or negative classifications. This type of information can help to determine the reliability of the test, as illustrated in Section 4.4.1 and Table 4.3.

If the value of Likelihood Ration Positive (**LRP**) is greater than 1, then the results indicate that they are associated with the positive class or presence of α-helix. If **LRP** is less than one, then the test is associated with the absence of the class. If **LRP** is greater than 5 then the pre-test probability can help to get the post-test probability. If the **LRP** is greater than 8, then it increases the likelihood of the predicted class actually belonging to that class. All these mea-

sures help determine the quality of the data representation in the model, gauge the reliability of the classification results with more confidence and can be used to fine tune error prone prediction regions in classifications. Next we discuss some of the other measures of accuracy commonly used in secondary structure predictions, which *assume a balanced representation* of all the classes during model building.

2.1.7.2 $\mathbf{Q_3}$ accuracy

$\mathbf{Q_3}$ is a commonly used measure for expressing the average accuracy for all three classes of secondary structures, while $\mathbf{Q_H}$, $\mathbf{Q_E}$ and $\mathbf{Q_C}$ are used to indicate individual accuracies for the three secondary structures. In the accuracy matrix [A_{ij}] of size 3 X 3, i and j correspond to the three classes H, E and C. The ijth element A_{ij} of the accuracy matrix is defined as the number of residues predicted to be in class j, which are actually observed to be in class i. The diagonal entries of [A_{ij}], are numbers of correctly predicted residues for each class where N is the total number of residues being classified. $\mathbf{Q_3}$ is defined as:

$$Q_3 = \frac{\sum A_{kk}}{N} \qquad where \quad i = j = k \tag{2.2}$$

The individual accuracies for each of the secondary structures, $\mathbf{Q_H}$, $\mathbf{Q_E}$ and $\mathbf{Q_C}$, are the percentage of correct predictions for each class with respect to the total number of samples present in each of those classes. If N is the total number of residues, and N_i are the residues in each secondary structure,

$$Q_i = \frac{A_{ii}}{N_i} \qquad where \quad i = H,\ E,\ C \tag{2.3}$$

2.1.7.3 Matthew's correlation coefficient

Matthew's correlation coefficient **(MCC)** is another commonly used measure to determine the quality of secondary structure predictions. It is defined as:

$$MCC_\alpha = \frac{TP\alpha * TN_\alpha - FN_\alpha * FP_\alpha}{\sqrt{([TN_\alpha + FN_\alpha]\,[TN_\alpha + FP_\alpha][TP_\alpha + FN_\alpha]\,[TP_\alpha + FP_\alpha])}} \tag{2.4}$$

MCC, for α-helix of class H as calculated above, is commonly used in machine learning studies to measure the quality of binary classifications. Although we use a multi-class (3-class) classification algorithm which simultaneously classifies all three classes, we calculate **MCC** by considering the positive classifications (α-helix) against the combined negative classifications (β-sheet and coil). **MCC** takes into account results of all positive and all negative classifications and is considered more balanced than other metrics like sensitivity or specificity which are not fair to all types of classification results as discussed earlier. This measure is important in protein secondary structure classifications where there is often an imbalance in the class representations among the three secondary structures. **MCC** shows the correlation between the observed and predicted values and ranges in value between +1 for perfect correlation and -1 for negative perfect correlation. An intermediate value means there is no correlation between observed and predicted values.

2.1.7.4 Segment Overlap score

Segment Overlap score (**SOV**) evaluates secondary structure predictions (**SSP**) on the basis of secondary structure segment overlaps rather than on the basis of accuracies with respect to individual residues in a sequence. **SOV** was defined in (Zemla et al., 1999). The traditional $\mathbf{Q_3}$ measure for **SSP** is not adequate for many purposes as discussed by those authors. $\mathbf{Q_3}$ scores for some secondary structure segments could show high accuracies while **SOV** scores can show them to be highly inaccurate. They argued that the type and location of secondary structures were very important for 3-D predictions and illustrated how $\mathbf{Q_3}$ accuracies can give an unrealistic, misleading and distorted assessment of the quality of **SSP**. They showed that **Q3** cannot differentiate between multiple breaks in a segment of secondary structure and a segment with only half the number of residues correctly predicted (without a break). A comprehensive discussion is given in their paper (Zemla et al., 1999). Hence, **SOV** has been increasingly considered as a more robust metric for estimating the quality of **SSP**. We use the

C-code for **SOV** shared by the authors, to obtain the **SOV** scores for our classifications.

2.1.7.5 $\mathbf{J}_1^{score}$ and $\mathbf{J}_2^{score}$

$\mathbf{J}_1^{score}$ and $\mathbf{J}_2^{score}$ are two coefficients that were proposed (Kloczkowski et al., 2002) as an improvement over **SOV**, to measure classification accuracy. The authors argued that even if one prediction had a larger number of residues predicted correctly compared to another prediction of the same segment, it would still have a lower **SOV** score if there was even a single residue misclassification in the middle of the segment, disrupting the predicted structure and causing a break. If another prediction of the sequence had a larger number of misclassified residues but with no breaks in the middle of the predicted secondary structure, that classification would have a higher **SOV**, although its accuracy score (total number of residues predicted correctly) would be lower. They also argued that **SOV** does not take relative positions of overlapping segments into account and does not credit predictions, which had structures more centered compared to other predictions. These are important considerations in using secondary structure predictions for 3-D modeling of proteins. $\mathbf{J}_1^{score}$ and $\mathbf{J}_2^{score}$ are much simpler to calculate compared to **SOV** calculations and were designed to overcome these problems. Within a given segment of secondary structure, each residue is assigned a weight, according to predetermined values, as defined in (Kloczkowski et al., 2002). The two scores differ only in the values of the weights that are assigned to the first four residues on either end of the segment. $\mathbf{J}_1^{score}$ and $\mathbf{J}_2^{score}$ will be included in the results of our future studies.

2.1.8 Limits of secondary structure predictability

Most of the successes in prediction studies were due to the availability of improved sequence alignment software such as **PSI-BLAST**, inclusion of non-homologous sequences, better **MSA** alignment programs such as **CLUSTAL-W** and the use of modern techniques such as machine learning algorithms and Hidden Markov Models (Karplus et al., 1997). Improvements in secondary structure prediction can originate in larger databases (Bairoch and Apweiler, 1997) or differences in secondary structure assignments. But many of these studies

have shown only a small 2% to 3% improvement from these factors. In a review of protein secondary structure prediction methods, Rost (Rost, 2001) suggested a theoretical limit of 88% for $\mathbf{Q_3}$ accuracy while more recent estimates (Pollastri et al., 2007) put this number between 90% and 95%. A recent study (Yang et al., 2011) using a two-level mixed-modal support vector machine, as discussed above, has obtained an accuracy of 85.6% using a combination of several methods and data. It is to be noted that prediction methods that use single protein sequences (without including **MSA** information) have not succeeded in predicting secondary structures beyond a $\mathbf{Q_3}$ accuracy of about 70%. Other studies that do use multiple sequence alignments have to resort to very complicated algorithms which might require large computational resources, time and expense to build the models and yet still yield accuracies mostly below 80%. Those that show considerable improvements use very complicated models, periods of up to three months for training the models and might need costly resources to achieve higher accuracy.

The Jernigan lab has proposed several secondary structure prediction methods recently and has set up the much improved **GOR V** server (Sen et al., 2005). The performance of **GOR V**, which includes evolutionary information from multiple sequence alignments, is presently comparable but slightly lower than the best cross-validated secondary structure prediction methods such as **PHD** (Rost, 1996, 2001) and **PSIPRED** (Jones, 1999). For example, the prediction accuracy measured by $\mathbf{Q_3}$ is 73.5% for GOR V, 71.9% for PHD, and 76.6% for an earlier version of **PSIPRED**. The Fragment Data Mining **(FDM)** algorithm for protein secondary structure prediction uses fragments of known structures obtained from multiple sequence alignment **(MSA)** of protein sequences. Its performance is excellent where high-scoring **MSA** matches are available. By combining the **FDM** with **GOR V**, a new Consensus Database Mining **(CDM)** method was developed (Cheng et al., 2007), which surpasses the performances of both **FDM** and **GOR V**. A web server for a Fragment Database Mining and Consensus Data Mining **(FDM/CDM)** approach has been set up by the Jernigan lab. This server has become more popular due to the reliability and efficiency of its performance, the simplicity of its use, and its potential for improvement with the rapidly growing number of determined

structures. Encouraged by the success of these methods, we have continued to seek for improvements in secondary structure prediction methods. We have incorporated information from increasingly available structure and sequence information in the protein databases as they become available. In the current study, in addition to using sequence information we incorporate long and short range interaction information for the amino acids in the sequences mined from existing structures, as discussed below.

2.1.9 Knowledge recovery from secondary structure predictions

Although machine learning methods have proved to be the most successful among other methods in secondary structure prediction, there has been a long standing objection to neural networks, since they are considered to be black boxes. It has been difficult to know the logic behind the process of classification to understand the accuracies are actually obtained. It would be good to know what factors or rules of classification contribute to these accuracies in order to understand the biological implications from such studies. A survey by (Tickle et al., 1998) revealed that several techniques have been developed (Andrews et al., 1995; Aldrich et al., 2000), which can help to extract a set of rules from neural network models. The strength of artificial neural networks comes from their ability to learn from the training of models. Using the knowledge gained from trained models, neural networks are then able to be generalized to unknown test cases. They achieve this goal by distributing their learning during modeling, in the form of weights and biases for the different neurons used in the modeling process. This capability to learn helps neural networks succeed where many other complicated algorithms and methods fail in real world applications.

Many researchers have worked with the problem of extracting knowledge from the workings of a neural network, but it has been a hard task due to the abstract nature of these networks. On the other hand, systems where rule-based symbolic languages such as Fuzzy Rule Bases **(FRBs)** are used are more comprehendible and these can be easily refined. Such symbolic rules could help to understand the decision principles that lead to the final decisions. Using these ideas, a hybrid intelligent system has been built (Kolman and Margaliot, 2005)

where symbolic and sub-symbolic information from the outputs of a neural network have been brought together. The synergy helped the authors to meld the capabilities of neural networks and the openness of **FRBs**. They reduced 64 parameters of the neuron network weights and biases down to 10 rules and were able to discern the knowledge behind the classification of numbers by a Light-Emitting Diode **(LED)** digital system in recognizing the 10 digits $0, 1, 2, ...9$ used by **LEDs**.

A similar approach can be used to extract the information from the neural networks used in secondary structure prediction studies. The gains will likely be significant but the task will be daunting. The **LED** system had only 64 weights (and biases) whereas the weights and biases in the neural network system that we have built number in the thousands. If computing resources are the only obstacles to get this information, this problem can easily be overcome, in so far as the complexity of the problem can be handled by powerful computers. It will be interesting to see such gains in knowledge in future studies.

2.2 Contribution of this thesis research to secondary structure prediction

The biggest contribution for our model is that it is *very simple, requires fewer resources and yields high accuracy* through a simple single layer neural network consisting only of *one such network*. The results from this algorithm are further optimized by using a very simple particle swarm optimization algorithm. These features make our algorithm highly efficient, extremely accurate and far less expensive to use compared to other proposed algorithms. The only drawback to this algorithm is in the time needed to generate the 27 features used to encode the sequence information for any new protein for which we need to predict the structure. It takes about a day to generate data for a small protein of less than 100 amino acids with reasonable resources on a single Linux machine while it might take up to a week for a bigger protein of 1500 residues, depending upon the capacity of the computer. With increasing availability of computer resources the data can be generated very quickly with an improved algorithm and a more powerful computer. The time and resources needed to build our training model is also much less compared to some algorithms, which can require a train-

ing period of several months. Our model needs a training period of only a few days to a week (computer dependent) after the actual training data has been generated and can be updated similarly when new information becomes available. Once the model is built it can be used for testing and it takes only a few seconds to determine the secondary structure for any new protein (not including the time needed for the data that needs to be generated for the new protein). Our studies yield results better than the virtual limit of 80%, and we have obtained much higher accuracies. We are able to predict secondary structures with a higher training accuracy of 93.33% and a testing accuracy of 92.24% on a group of 84 proteins, which shows excellent generalization performance. Breaking it down, the contributions for the high accuracies come from the high individual testing accuracies of 94.19% for α-helix, 92.39% for β-strand and 91.11% for coil, resulting in very low standard deviations, ranging from 0.3% to 2.78% for the 20 types of amino acids. We have a Matthew's correlation-coefficient ranging between 80.58% and 84.30% for these secondary structures. On a larger set of 415 proteins, we obtained a testing accuracy of 86.5%. These results are significantly better than those found in the literature even if compared with studies that include **MSAs** , while our studies *use only sequence and existing structure information* from the **CATH** (Orengo et al., 1997) database.

2.2.1 ELM-PSO for secondary structure prediction

Initially, a novel method called **ELM-PSO** for predicting protein secondary structure, using data derived from knowledge-based potentials and an Extreme Learning Machine, was developed (Saraswathi et al., 2010b). Classifier performance was maximized using the generally available Particle Swarm Optimization (**PSO**) algorithm. Preliminary results for **ELM-PSO** were good when prior information was used in the form of scaled feature values. Since prior information will not be available for newer protein sequences, other methods were investigated. **ELM-PSO** was improved to exclude prior scaling information. A new model called **FLOPRED** was developed which used advanced **PSO** algorithms to give higher classification accuracies.

2.2.2 FLOPRED for secondary structure prediction

In our current studies, a novel method called **Fast Learning Optimized Predictor (FLOPRE)** has been proposed for predicting protein secondary structure and protein function, using knowledge-based potentials, Neural Networks and Particle Swarm Optimization. Higher secondary structure prediction accuracies are achieved by applying the FLOPRED algorithm to the **CB513** (Cuff and Barton, 2000) set of proteins which are the *target sequences* for which we need to predict secondary structures. While there are many more recent datasets, we used the **CB513** set in order to be able to compare our results with several others present in the literature. We plan to test the **FLOPRED** algorithm on a newer larger set of proteins also. The protein sequences in the dataset were encoded with long-range and short-range interactions, using potentials extracted by using **CABS** (Kolinski, 2004) algorithm. Details of the data generation are given in Section 2.3 on page 46.

We use a machine learning method called Extreme Learning Machine **(ELM)** (Huang et al., 2006). This algorithm is based on a traditional Neural Network (NN) and can be used for classification of protein data, such as sequences and other related information. An Extreme Learning Machine **(ELM)** classifier, based on a Neural Network, is used to model and predict protein secondary structure. ELM is an improved version of a feed-forward neural network consisting of a single hidden layer. The initial set of input weights is chosen randomly. The output weights from the hidden layer to the output layer are calculated analytically. A sigmoidal (or Gaussian) activation function is used for the hidden layer and a linear activation function is used for the output neurons.

In initial studies, the input weights and other parameters of **ELM** were tuned using a simpler Particle Swarm Optimization **(PSO)** algorithm (Kennedy and Eberhart, 1995). In our current studies, an improved and extended family of advanced **(PSO)** algorithms (Fernández-Martínez and García-Gonzalo, 2008, 2009, 2010) have been used to tune the parameters (hidden neurons, bias and width) of the sigmoidal/Gaussian activation function of **ELM**. The use of these efficient algorithms has resulted in much improved accuracies for all predictions as discussed in Section 2.2 on page 42. These algorithms are explained in methods in Sec-

tion 1.4.1 on page 6 and under optimization in Section 1.4.2 on page 11.

2.2.3 An amino acid perspective of secondary structure prediction

For many years, researchers have worked with protein structure prediction and they have had highly varied degrees of success while slow progress has been made beyond particular thresholds. There is a need for *improved structure and functional-site prediction methods* to increase *accuracy and efficiency*, in view of the need to predict the structures of millions of sequences. It is of interest to *analyze the reasons* for being unable to predict secondary structures, beyond a particular degree of accuracy. While there might be several reasons for this, we offer an amino acid perspective of the prediction results in order to investigate the nature of secondary structure prediction with respect to amino acid composition and ease of prediction. This analysis throws some light on the nature of weakly predicted regions (regions where errors occur more frequently) in protein sequences with respect to the amino acid compositions. The results of this study is given in Section 5.1.1 on page 101 .

2.2.4 Use of physicochemical properties for secondary structure prediction

In order to determine the effects of biophysical properties of amino acids on formation of particular secondary structures, a database was set up where protein sequences from the **CB513** data set were encoded using 544 physicochemical properties of amino acids derived from the **AAindex** (Kawashima et al., 1999) database. A window of 9 residues was used to code the 544 properties which resulted in 4896 features. These features were reduced to less than 150 features using Genetic Algorithm **(GA)** 6.2.2 described on page 118 and Principal Component Analysis **(PCA)** 6.2.4 described on page 123. This reduced dataset was then used for secondary structure prediction using our **FLOPRED** algorithm. Preliminary results show 82% accuracy for training and 65% for testing. These results are discussed in Section 6.3 on page 125.

2.2.5 Use of position specific propensities of amino acids for secondary structure prediction

Due to physicochemical properties, some amino acids appear more often at the ends of secondary structures than others. A preliminary study has indicated that secondary structure accuracy can be improved for those residues present at the ends of α-helix, β-strand and coil. Hence, information on Position Specific Residue Preferences (**PSRP**) of amino acids, can be used to improve secondary structure predictions. Preliminary studies show that **PSRP** values can contribute as little as 6% or as much as 15%, depending on the models used for representing the ends of secondary structures. These results are discussed in Section 7.6 on page 139. We hope to use the **PSRP** information as prior knowledge to improve secondary structure prediction results in future studies.

2.3 Data generation using CABS force field

The **CB513** data set (Cuff and Barton, 2000) is a collection of a set of 513 non-redundant protein domains that has less than 30% homology between the pairs of sequences. This dataset is used for all the secondary structure prediction studies in our research. The protein sequences in the **CB513** set are *the target sequences* that are used for both model building and testing purposes.

Data derived from the potential energies of amino acids in the **CB513** set of protein sequences were encoded into three secondary structures using the **CABS** force field (Kolinski, 2004). The secondary structure assignments were discussed earlier in Section 2.1.5 on page 30. **CABS** is a "versatile reduced representation tool for molecular modeling" (Kolinski, 2004). This algorithm encodes both short-range and long-range interactions in proteins. **CABS** stands for C-α-C-β-Side group protein model where C-α is the α-carbon and C-β is the β-carbon in an amino acid backbone structure. This algorithm uses a high resolution reduced model of proteins and the force field. It uses a lattice model to represent hundreds of possible orientations of the virtual α-carbon-α-carbon bonds, using Replica Exchange Monte Carlo for sampling the conformational space. The knowledge-based potentials of the force

field includes the following information:

- Protein-like conformational biases
- Statistical potentials for the short-range interactions
- A representation of main chain hydrogen bonds
- Statistical potentials describing the side chain interactions.

The **CABS** model is an accurate lattice model and has been used in many applications to represent proteins in a reduced representation. Our knowledge-based potential data generation consists of the following steps:

- Download templates from the **CATH** (Cuff et al., 2008) database.
- Compute secondary structure information using **DSSP** for each residue in each template.
- Compute contact maps for each template, including both secondary and tertiary interactions.
- Thread a window of 17 sequences for each template sequence, onto each of the 422 templates and calculate the reference energy for each residue in *all templates*. A list of these templates is given in Appendix A.
- Thread a window of 17 residues for each of the target sequences onto each template and calculate the reference energy for each residue in *all target sequences*.
- Read in the **DSSP** information for the window of residues for the template sequences which has the best fit. This is done only for the central 9 residues in each window.
- Find the probability that the 9 residues in the window will adapt to *each* of the three secondary structures, to obtain 27 feature values.

2.3.1 Structures from the CATH database

A database of protein structure templates has been created by downloading structures from the **CATH** (Orengo et al., 1997) library. We are interested in the connectivity between the residues in the sequence fragments under consideration. Hence the structures found at the topology level were downloaded since these have the same overall fold (Orengo et al., 1997; Cuff et al., 2008), and share similarity in the arrangement and interconnectivity of structural elements. For each structure template, secondary structure information was found using the **DSSP** algorithm (Kabsch and Sander, 1983). Only those templates with complete information were downloaded from CATH in order to ensure accurate and complete **DSSP** secondary structure assignments for a given sequence fragment. This method of selection yielded 422 structures.

2.3.2 Contact maps and reference energy for template sequences

A contact map has been computed for each of the 422 templates, with two residues considered to be in contact, if the distance between their heavy atoms (carbon, oxygen and nitrogen) is less than 4.5 Å. The contact maps are used to calculate the force field values.

A window of 17 amino acids is used where the 9th (central) amino acid is the residue of interest, with null spaces representing places where no neighbors are available at the starts and ends of sequences. The window is moved, one residue at a time, along the full length of the sequence. Each of these windows in each sequence is considered for each of the 422 templates, by placing the window centered on the 9^{th} residue. The sequence fragment inside the window adopts the structure of the template where it is placed. A reference energy is then calculated for each amino acid in each of the 422 templates.

2.3.3 Reference energy for the target sequences

The **CB513** dataset is used for potential energy extraction. **CB513**, a collection of non-redundant protein domains (Cuff and Barton, 2000) with less than 30% homologous sequences

has been used for the target sequences. Reference energy is calculated for the target sequences using a non-gapped threading procedure with the 422 template structures.

2.3.4 Threading procedure for calculating reference energy

The template structures are used to search for a match with the residues in the window. When a match is found a scoring function (unpublished) is used to assess and calculate the degree of compatibility. For each of these placements,the secondary and tertiary energy is calculated and the lowest energy values are retained. For example, for the fourth amino acid in a target sequence, we might have obtained the lowest energy (best fit), while it was centered on the 10th amino acid of a template sequence.

2.3.5 Secondary structure assignment and creation of profile matrices

The secondary structure assignments from **DSSP** (Kabsch and Sander, 1983) are read in for the template sequences for which the best fit was determined. Although the window originally consisted of 17 residues, only the values for the central 9 residues are utilized henceforth, for each of the three secondary structures, α-helix, β-sheet and coil. The final profile matrix, consists of one row of data for each of the residues represented by the sequence of a given protein. Each row has a set of 27 features (profile values), where the first 9 features correspond to the probability that the residues from the target sequence, adopt an α-helix (H) structure. The next 9 features, correspond to the probability that they adopt an extended β-strand (E) and the last 9 features correspond to the probability that they adopt a coil (C) structure. The probability p of getting such a threading match is then determined (Silva, 2008).

2.3.6 Calculation of reference energy

Reference energy are calculated using the (Kolinski, 2004) **CABS** force fields. Short range, long range and hydrophobic sequence dependent interactions are calculated. R13, R14 and R15 potentials depend on the geometry and identity between the i^{th} and i+2^{nd}, i+3^{rd} and i+4^{th} amino acids respectively. Sequence dependent (short-range) interactions for these residues

are calculated. In order to include long-range interactions, a contact energy is added to the previously calculated energy values only for the aligned residues observed to be in contact after the threading procedure has been done. The contact information comes from the contact maps established for each template. A score for the hydrophobic and hydrophilic amino acid matches between the template and target sequence fragments is also calculated (Silva, 2008). The energy values from these three calculations are weighted in the ratio $2.0 : 0.5 : 0.8$ for the long : short : hydrophobic interactions respectively. The selected weights are based on other computations for 3-D threading (unpublished), although it has been found that the results are not very sensitive to the selection of these parameters.

2.3.7 Homology between template and target sequences

Since the energy profiles are based on the template sequences, we need to make sure that a low degree of homology exists between the 422 **CATH** template structures and **CB**513 set of target sequences. A global Needleman-Wunsch (Needleman and Wunsch, 1970) sequence alignment was performed using the BLOSUM62 (Henikoff and Henikoff, 1992) matrix, with a penalty function of 10 for an initial gap and 1 for gap extensions. Figure 2.1 shows a histogram of the similarity scores for the approximately $500,000$ pair-wise sequence alignments, between the 1000 templates (initially selected) and the 513 target sequences. Although, initially 1000 structures were selected from the CATH database, subsequently only 422 of these templates were used (due to errors in the PDB files and computational resource concerns). For the set of 513 proteins, those which were found to have more than 70% sequence similarity with at least one of the 1000 templates have been removed. These proteins were not included during cross-validation or testing of the data. For this reason, the final data set was reduced to 415 proteins which has less than 70% homology with the template sequences. As shown in Figure 2.1 on page 61 the overall similarity is very less for all the $513 * 1000$ pairwise alignments. It can be seen that 97% of the global-Needleman-Wunch pair-wise alignments (Needleman and Wunsch, 1970) have between 10% and 18% homology between the template and the target sequences. The list of 422 templates are given in Appendix A on page 178.

Individual homology scores have not been shown.

2.3.8 Homology between template and target structures

We performed a structure comparison study using Homology-derived Secondary Structure of Proteins (**HSSP**) (Sander and Schneider, 1991) in order to detect structure similarity between the CB513 target set and the template sequences. This was done to eliminate the possibility that structural similarity might contribute to the higher accuracy that we obtain from the initial study. We downloaded all of the 422 template files that were used for data generation from **RSCB PDB**. Then we searched for the name of each of the 513 proteins in the downloaded **HSSP** files to see if any of the H**SSP** files contain any of target proteins as recognized structure similar to the templates. The results show that there were only 23 proteins having structures similar to the templates. Of these only 3 were included in the initial study using dataset-84 and 23 were included in the final study using dataset-415. These will removed from our final results before submitting the paper. This test was done recently to make sure that we have not included any structure information in our data in order to get higher accuracies. Considering the very low standard deviations for the results which range from 0.3% to 2.78% for all the amino acids, removal of these 23 proteins from the data set is not expected to have a huge impact on the results for either set. The results are still expected to be above what is seen in the literature.

2.4 Summary of secondary structure studies conducted in this thesis

In summary, the 27 profiles for each of the amino acids in each of the 415 target sequences have been calculated using the **CABS** force field (Kolinski, 2004). This data was used for model building and testing of classification accuracy. The data generation process is very computationally intense and is dependent on the number of template structures used for threading and the size of the target protein set. It took several hours for a small protein (100 to 300 residues) and several days for a large protein (over 1000 residues), depending on the computers used. Traditionally, orthogonal binary representations and **PSSM** (Jones, 1999)

profile matrices (which are easily generated) are used to represent amino acids in protein sequences. Since the energy calculations using the **CABS** algorithm are very computationally intensive, the time involved in generating the profile matrices can be a limiting factor in using our algorithm.

REFERENCES

Aldrich, C., Cervenka, J., Cloete, I., Cozzio, R. A., Drossu, R., Fletcher, J., Giles, C. L., Gouws, F. S., Hilario, M., Ishikawa, M., Lozowski, A., Obradovic, Z., Omlin, C. W., Riedmiller, M., Romero, P., Schmitz, G. P. J., Sima, J., Sperduti, A., Spott, M., Weisbrod, J., and Zurada, J. M. (2000). Knowledge-based neurocomputing. In Cloete, I. and Zurada, J., editors, *Knowledge-Based Neurocomputing*. MIT Press.

Altman, D. G. and Bland, J. M. (1994). Statistics notes: Diagnostic tests 2: predictive values. *BMJ*, 309:102.

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST a new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402.

Andrews, R., Diederich, J., and Tickle, A. (1995). Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Systems*, 8:373–389.

Artymiuk, P., Taylor, W., and Phillips, D. (2011). Triose phosphate isomerase. *To be Published*, DOI:10.2210/pdb8tim/pdb.

Babaei, S., Geranmayeh, A., and Seyyedsalehi, S. A. (2010). Protein secondary structure prediction using modular reciprocal bidirectional recurrent neural networks. *Computer Methods and Programs in Biomedicine*, 100:237–247.

Bairoch, A. and Apweiler, R. (1997). The SWISS-PROT protein sequence database: its relevance to human molecular medical research. *Journal of Molecular Medicine*, 75:312–316.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000). The protein data bank. *Nucleic Acids Research*, 28:235–242.

Bidargaddi, N. P., Chetty, M., and Kamruzzaman, J. (2009). Combining segmental semi-Markov models with neural networks for protein secondary structure prediction. *Neurocomputing*, 72:3943–3950.

Cheng, H., Sen, T. Z., Jernigan, R. L., and Kloczkowski, A. (2007). Consensus Data Mining (CDM) protein secondary structure prediction server: combining GOR V and Fragment Database Mining (FDM). *Bioinformatics*, 23:2628–2630.

Chou, P. Y. and Fasman, G. D. (1974). Prediction of protein conformation. *Biochemistry*, 13:222–245.

Cuff, A. L., Sillitoe, I., Lewis, T., Redfern, O. C., Garratt, R., Thornton, J., and Orengo, C. A. (2008). The CATH classification revisited-architectures reviewed and new ways to characterize structural divergence in superfamilies. *Nucleic Acids Research*, 37:D310–314.

Cuff, J. A. and Barton, G. J. (2000). Application of multiple sequence alignment profiles to improve protein secondary structure prediction. *Proteins*, 40:502–511.

Eddy, S. R. (1998). Profile hidden markov models. *Bioinformatics*, 14:755–763.

Fernández-Martínez, J. L. and García-Gonzalo, E. (2008). The Generalized PSO: A New Door to PSO Evolution. *Journal of Artificial Evolution and Applications*, 2008:15.

Fernández-Martínez, J. L. and García-Gonzalo, E. (2009). The PSO family: deduction, stochastic analysis and comparison. *Special issue on PSO. Swarm Intelligence*, 3:245–273.

Fernández-Martínez, J. L. and García-Gonzalo, E. (2010). Two algorithms of the extended PSO family. In *Proceedings of IJCCI/ICNC*, pages 237–242.

Frishman, D. and Argos, P. (1995). Knowledge-based protein secondary structure assignment. *Proteins*, 23:566–579.

Garnier, J., Gibrat, J. F., and Robson, B. (1996). GOR secondary structure prediction method version IV. *Methods in Enzymology*, 226:540–553.

Garnier, J., Osguthorpe, D. J., and Robson, B. (1978). Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *Journal of Molecular Biology*, 1:97–120.

Ghosh, A. and Parai, B. (2008). Protein secondary structure prediction using distance based classifiers. *International Journal of Approximate Reasoning*, 47:37–44.

Henikoff, S. and Henikoff, J. (1992). Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences U.S.A.*, 89:10915–10919.

Huang, G. B., Zhu, Q. Y., and K, S. C. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, 70:489–501.

Humphrey, W., Dalke, A., and Schulten, K. (1996). VMD - visual molecular dynamics. *Journal of Molecular Graphics*, 14:33–38.

Jones, D. (1999). Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology*, 292:195–202.

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

Karplus, K., Sjölander, K., Barrett, C., Cline, M., Haussler, D., Hughey, R., Holm, L., and Sander, C. (1997). Predicting protein structure using hidden markov models. *Proteins*, Suppl 1:134–139.

Kashlan, O. B., Maarouf, A. B., Kussius, C., Denshaw, R. M., Blumenthal, K. M., and Kleyman, T. R. (2006). Distinct structural elements in the first membrane-spanning segment of the epithelial sodium channel. *Journal of Biological Chemistry*, 281:30455–30462.

Kawashima, S., Ogata, H., and Kanehisa, M. (1999). AAindex: Amino Acid Index Database. *Nucleic Acids Research*, 27:368–369.

Kazemian, M., Moshiri, B., Nikbakht, H., and Lucas, C. (2007). A new expertness index for assessment of secondary structure prediction engines. *Computational Biology and Chemistry*, 31:44–47.

Kennedy, J. and Eberhart, R. C. (1995). Particle Swarm Optimization. *Proceedings of the IEEE International Conference on Neural Networks*, 4:1942–1948.

Kihara, D. (2005). The effect of long-range interactions on the secondary structure formation of proteins. *Protein Science*, 14:1955–1963.

Kloczkowski, A., Ting, K. L., Jernigan, R. L., and Garnier, J. (2002). Combining the GOR V algorithm with evolutionary information for protein secondary structure prediction from amino acid sequence. *Proteins*, 49:154–166.

Kolinski, A. (2004). Protein modeling and structure prediction with a reduced representation. *Acta Biochem Pol.*, 51:349–371.

Kolman, E. and Margaliot, M. (2005). Knowledge extraction from neural networks using the all-permutations fuzzy rule base: The led display recognition problem. In *Computational Intelligence and Bioinspired Systems*, volume 3512, pages 88–113. Springer Berlin / Heidelberg.

Krissinel, E. and Henrick, K. (2004). Secondary structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallogr D Biol Crystallogr*, 60:2256–2268.

Liu, N. and Wang, T. (2007). A simple method for protein structural classification. *Journal of Molecular Graphics and Modeling*, 25:852–855.

Lomize, A. L., Pogozheva, I. D., and Mosberg, H. I. (1999). Prediction of protein structure : The problem of fold multiplicity. *Proteins*, 37:199–203.

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

Malekpour, S. A., Naghizadeh, S., Pezeshk, H., Sadeghi, M., and Eslahchi, C. (2009). Protein secondary structure prediction using three neural networks and a segmental semi markov model. *Mathematical Biosciences*, 217:145–150.

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

Montgomerie, S., Sundaraj, S., Gallin, W., , and Wishart, D. (2006). Improving the accuracy of protein secondary structure prediction using structural alignment. *BMC Bioinformatics*, 301:301.

Needleman, S. B. and 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*, 48:443–453.

Oklejas, V., Zong, C., Papoian, G. A., and Wolynes, P. G. (2010). Protein structure prediction: Do hydrogen bonding and water-mediated interactions suffice? *Methods*, 52:84–90. Protein Folding.

Orengo, C. A., Michie, A. D., Jones, D. T., and Swindells, M. B. andThornton, J. M. (1997). Cath: A hierarchic classification of protein domain structures. *Structure*, 5:1093–1108.

Ortiz, A. R., Kolinski, A., Rotkiewicz, P., Ilkowski, B., and Skolnick, J. (1999). Ab initio folding of proteins using restraints derived from evolutionary information. *Proteins*, 37:177–185.

Palopoli, L., Rombo, S. E., Terracina, G., Tradigo, G., and Veltri, P. (2009). Improving protein secondary structure predictions by prediction fusion. *Information Fusion*, 10:217–232. Special Issue on Natural Computing Methods in Bioinformatics.

Paoli, M., Liddington, R., Tame, J., Wilkinson, A., and G., D. (1996). Crystal structure of T state haemoglobin with oxygen bound at all four haems. *Journal of Molecular Biology*, 256:775–792.

Pollastri, G., Martin, A., Mooney, C., and Vullo, A. (2007a). Accurate prediction of protein secondary structure and solvent accessibility by consensus combiners of sequence and structure information. *BMC Bioinformatics*, 8:201.

Pollastri, G., Martin, A., Mooney, C., and Vullo, A. (2007b). Accurate prediction of protein secondary structure and solvent accessibility by consensus combiners of sequence and structure information. *BMC Bioinformatics*, 8:201.

Pruitt, K. D., Tatusova, T., Klimke, W., and Maglott, D. R. (2009). NCBI reference sequences: current status, policy and new initiatives. *Nucleic Acids Research, Database issue*, 37:D32–D36.

Qian, N. and Sejnowski, T. J. (1988). Predicting the secondary structure of globular proteins using neural network models. *Journal of Molecular Biology*, 202:865–884.

Rost, B. (1996). Phd: predicting one-dimensional protein structure by profile-based neural networks. *Methods in Enzymology*, 266:525–539.

Rost, B. (2001). Review: Protein secondary structure prediction continues to rise. *Journal of Structural Biology*, 134:204–218.

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

Rost, B., Yachdav, G., and Liu, J. (2004). The predictprotein server. *Nucleic Acids Research*, 32:W321–W326.

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

Salamov, A. A.and Solovyev, V. V. (1997). Protein secondary structure prediction using local alignments. *Journal of Molecular Biology*, 268:31–36.

Salzberg, S. and Cost, S. (1992). Predicting protein secondary structure with a nearest-neighbor algorithm. *Journal of Molecular Biology*, 227:371–374.

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

Santiago-Gómez, M. P., Kermasha, S., Nicaud, J.-M., Belin, J.-M., and Husson, F. (2010). Predicted secondary structure of hydroperoxide lyase from green bell pepper cloned in the yeast *Yarrowia lipolytica*. *Journal of Molecular Catalysis B: Enzymatic*, 65:63–67.

Saraswathi, S., Jernigan, R. L., Koliniski, A., and Kloczkowski, A. (2010). Protein secondary structure prediction using knowledge-based potentials. *Proceedings of IJCCI/ICNC*, pages 370–375.

Sen, T. Z., Jernigan, R. L., Garnier, J., and Kloczkowski, A. (2005). GOR V server for protein secondary structure prediction. *Bioinformatics*, 21:2787–2788.

Silva, P. J. (2008). Assessing the reliability of sequence similarities detected through hydrophilic cluster analysis. *Proteins*, 70:1588.

Sivan, S., Filo, O., and Siegelmann, H. (2007). Application of expert networks for predicting proteins secondary structure. *Biomolecular Engineering*, 24:237–243.

Tickle, A. B., Andrews, R., Golea, M., and Diederich, J. (1998). The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural networks. *IEEE Trans. on Neural Networks*, 9:1057–1068.

Vapnik, V. N. (2000). *The Nature of Statistical Learning Theory (Information Science and Statistics)*. Springer-Verlag, New York.

Vijay-Kumar, S., Bugg, C., and Cook, W. (1987). Structure of ubiquitin refined at 1.8 åresolution. *Journal of Molecular Biology*, 194:531–544.

Wang, G., Zhao, Y., and Wang, D. (2008). A protein secondary structure prediction framework based on the extreme learning machine. *Neurocomputing*, 72:262–268.

Ward, J. J., McGuffin, L. J., Buxton, B. F., and Jones, D. T. (2003). Secondary structure prediction with support vector machines. *Bioinformatics*, 19:1650–1655.

Wray, L. V. and Fisher, S. (2007). Functional analysis of the carboxy-terminal region of bacillus subtilis tnra, a merr family protein. *Acta Crystallogr D Biol Crystallog*, 189:20–27.

Yang, B., Wei, H., Zhun, Z., and Huabin, Q. (2009). KAAPRO An approach of protein secondary structure prediction based on KDD* in the compound pyramid prediction model. *Expert Systems with Applications*, 36:9000–9006.

Yang, B., Wu, Q., Ying, Z., and H., S. (2011). Predicting protein secondary structure using a mixed-modal SVM method in a compound pyramid model. *Knowledge-Based Systems*, 24:304–313.

Yi, T. M. and Lander, E. S. (1993). Protein secondary structure prediction using nearest-neighbor methods. *Journal of Molecular Biology*, 232:1117–1129.

Yuksektepea, F. U., Yilmaz, O., and Türkay, M. (2008). Prediction of secondary structures of proteins using a two-stage method. *Computers & Chemical Engineering*, 32:78–88.

Zemla, A., Venclovas, e., Fidelis, K., and Rost, B. (1999). A modified definition of SOV, a segment-based measure for protein secondary structure prediction assessment. *Proteins: Structure, Function, and Bioinformatics*, 34:220–223.

Zhang, S., Ding, S., and Wang, T. (2011). High-accuracy prediction of protein structural class for low-similarity sequences based on predicted secondary structure. *Biochimie*, 93:710–714.

Zhou, Z., Yang, B., and Hou, W. (2010). Association classification algorithm based on structure sequence in protein secondary structure prediction. *Expert Systems with Applications*, 37:6381–6389.

Zvelebil, M. J., Barton, G. J., Taylor, W. R., and Sternberg, M. J. E. (1987). Prediction of protein secondary structure and active sites using the alignment of homologous sequences. *Journal of Molecular Biology*, 195:957–961.

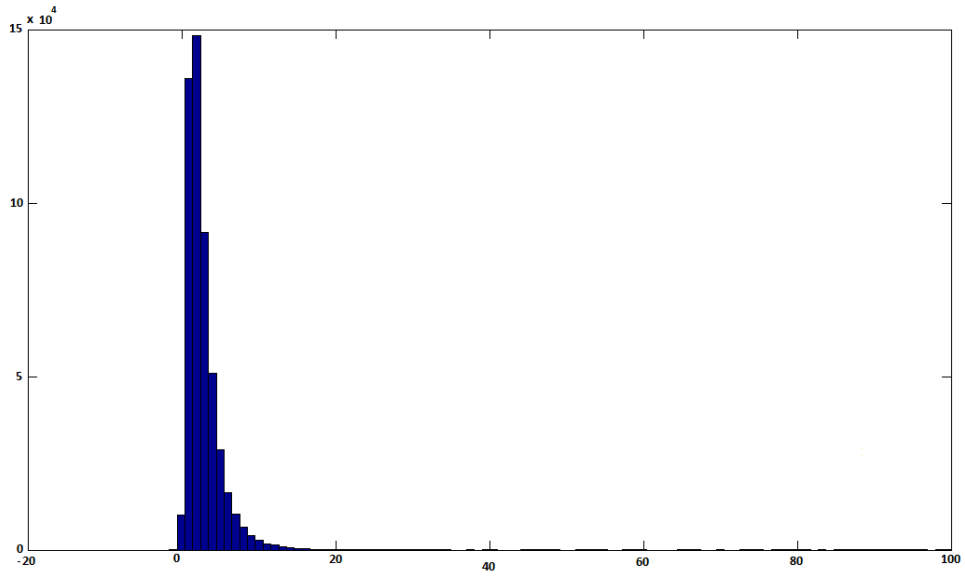


Figure 2.1 Sequence homology between templates and the set of 513 target sequences

This figure shows the sequence homology between the 1000 template sequences and the set of 513 target sequences. There are a total of $513 * 1000 = 513000$ pair-wise alignments which gave scores of between 0% and 99% homology between pairs. The x-axis shows the % sequence similarity scores. There were 499175 pair-wise alignments with less than 10% sequence similarity and 12480 sequences with between 10% and 20% similarity. The remaining 8 bins had $308, 14, 14, 6, 8, 9, 25$ and 37 pair wise sequence similarities for intervals in the range of $30, 40, 50, 60, 70, 80, 90, 100\%$ where each bin will be between 30% and 40% etc. The number of pair-wise sequences with over 70% homology were $37 + 25 + 9 = 71$ sequences which were removed from the study. It can be seen that 97% of the alignments that were preformed, using the global-Needleman-Wunch (Needleman and Wunsch, 1970) algorithm, have less than 10% homology with the templates target sequences.

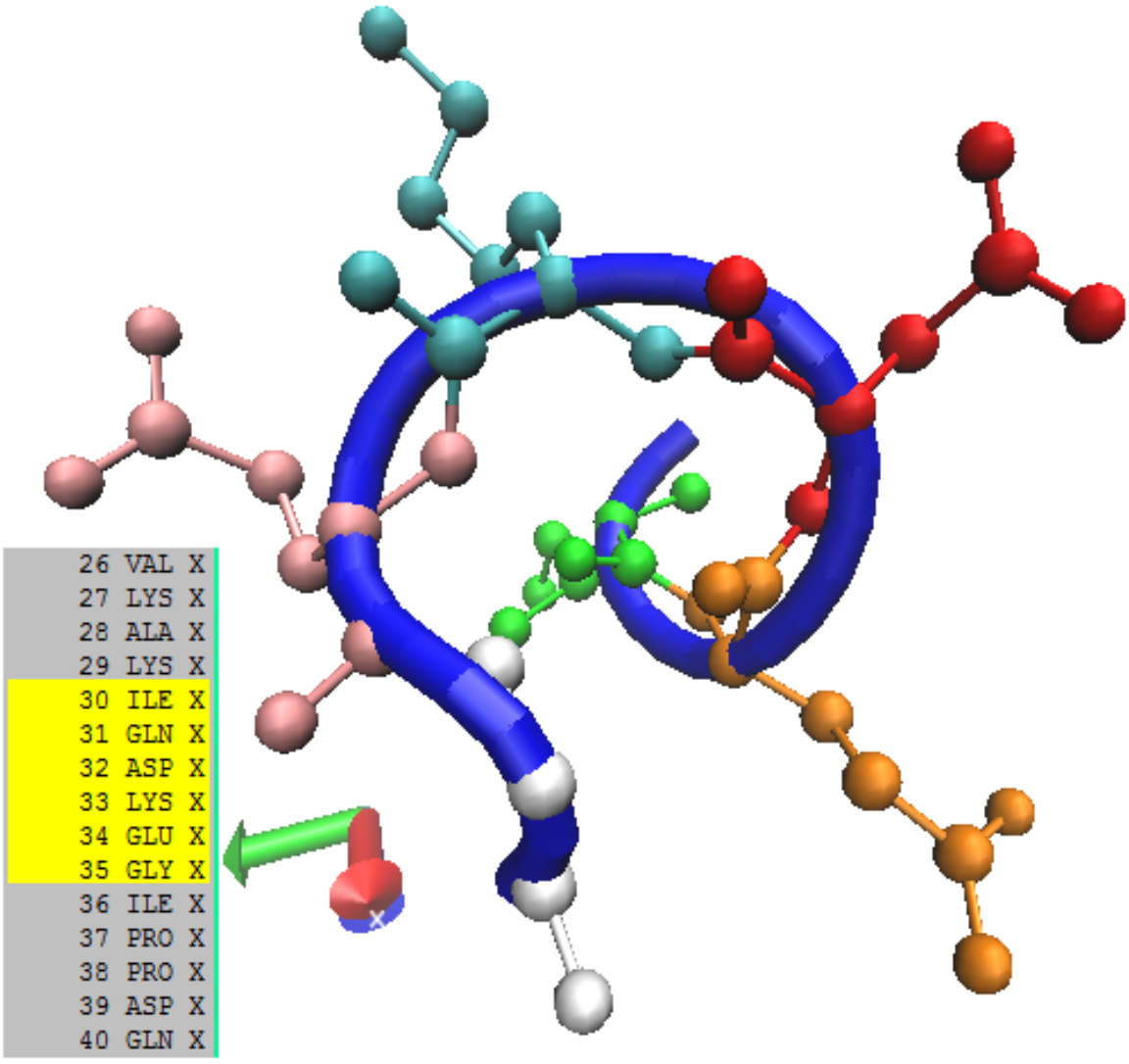


Figure 2.2 Proteins and their amino acids

This figure shows residues 30 to 35 from the ubiquitin protein (1ubq.pdb) (Berman et al., 2000; Vijay-Kumar et al., 1987). The residues shown are marked in yellow in the list shown and are ordered anti-clockwise starting with isoleucine in green and ending in glycine in white. These residues were rendered using VMD (Humphrey et al., 1996).

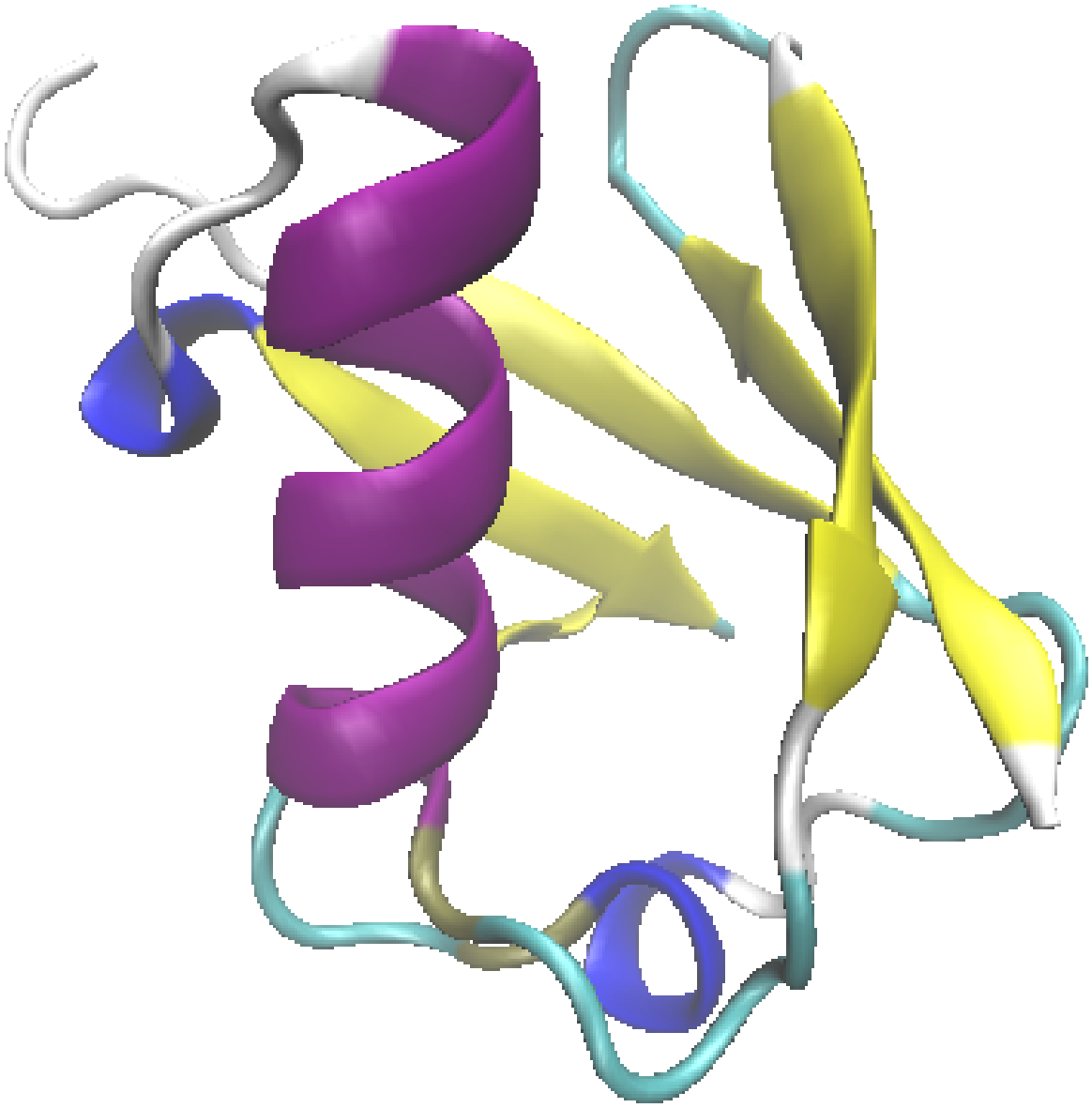


Figure 2.3 Proteins and their secondary structures

This figure shows the secondary structures of Ubiquitin protein (1ubq.pdb) (Berman et al., 2000; Vijay-Kumar et al., 1987) which has three and one half turns of α-helix (purple), one short piece of 3_{10}-helix and a mixed β-sheet (yellow) with five strands and seven reverse turns (light blue). These secondary structures were computed and assigned according to the STRIDE (Frishman and Argos, 1995) classification of secondary structures. The residues were rendered using VMD (Humphrey et al., 1996).

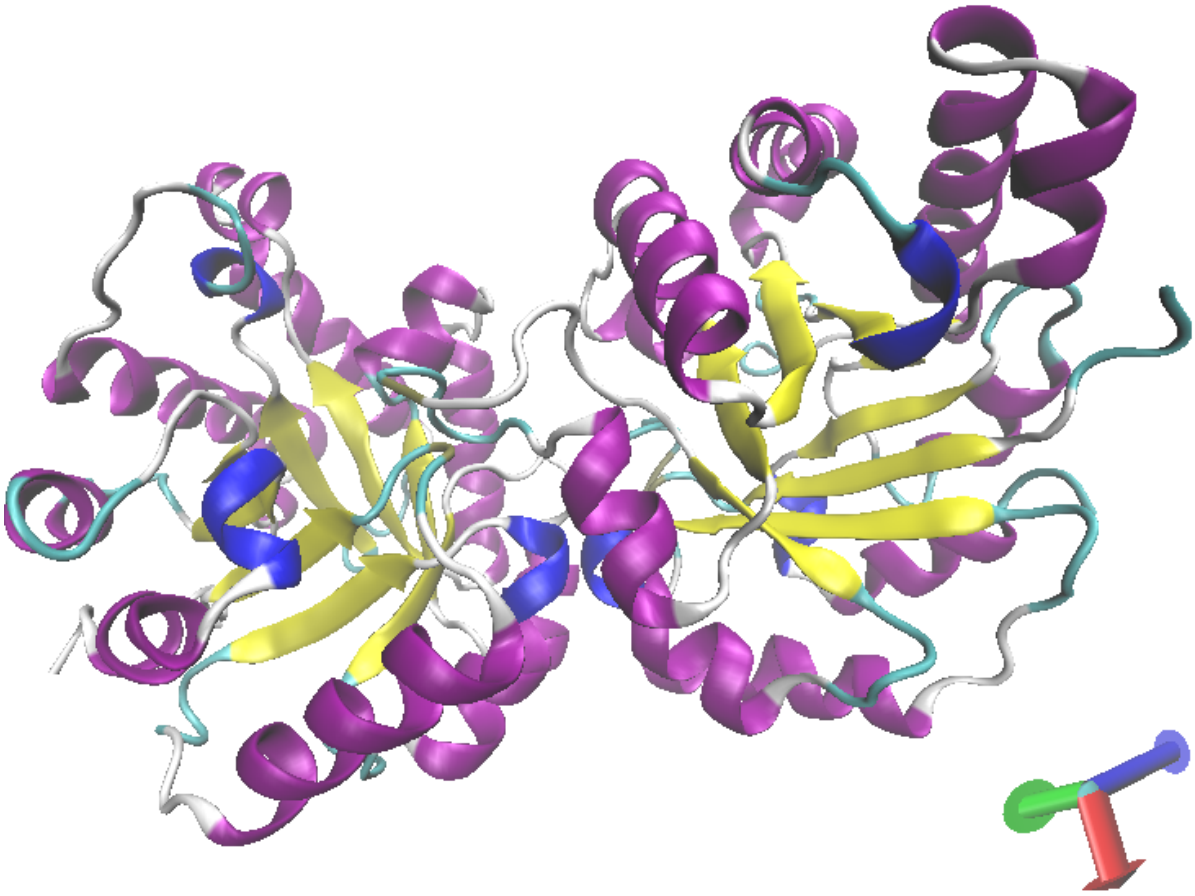


Figure 2.4 Proteins and their tertiary structures

This figure shows the TIM barrel fold of protein (8TIM.pdb) (Berman et al., 2000; Artymiuk et al., 2011) triose phosphate isomerase, an enzyme. This protein has 8 α-helices (purple), and 8 β-sheets (yellow) in an alternating pattern, in each domain. This protein was rendered using VMD (Humphrey et al., 1996).

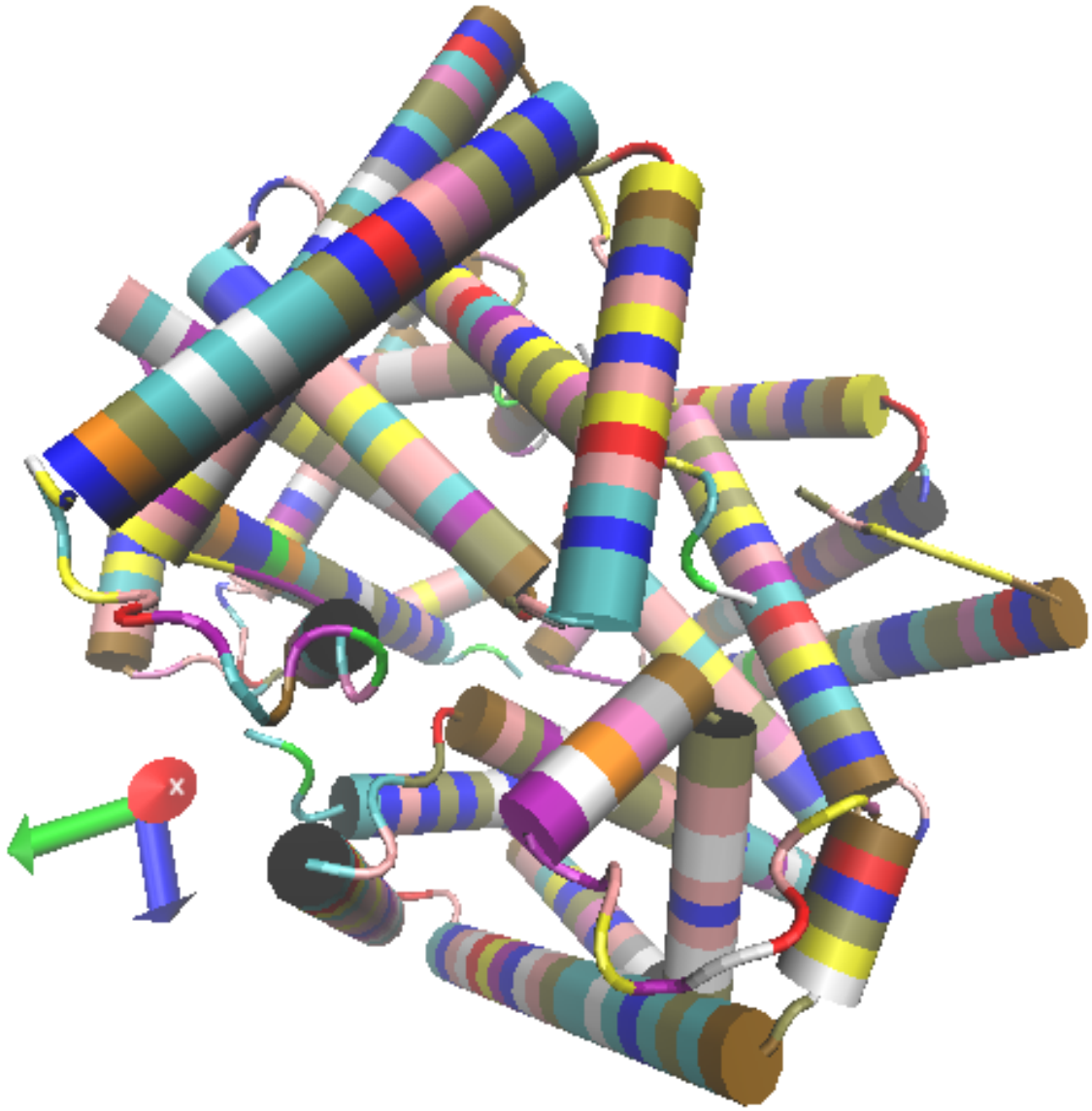


Figure 2.5 Proteins and their quaternary structures

This figure shows the quaternary structure of hemoglobin (1GZX.pdb) (Berman et al., 2000; Paoli et al., 1996) which has four protein chains, two α chains and two β chains, colored according to the residue type. It has a heme group, which binds oxygen atom(not shown). The structure was rendered using VMD (Humphrey et al., 1996).

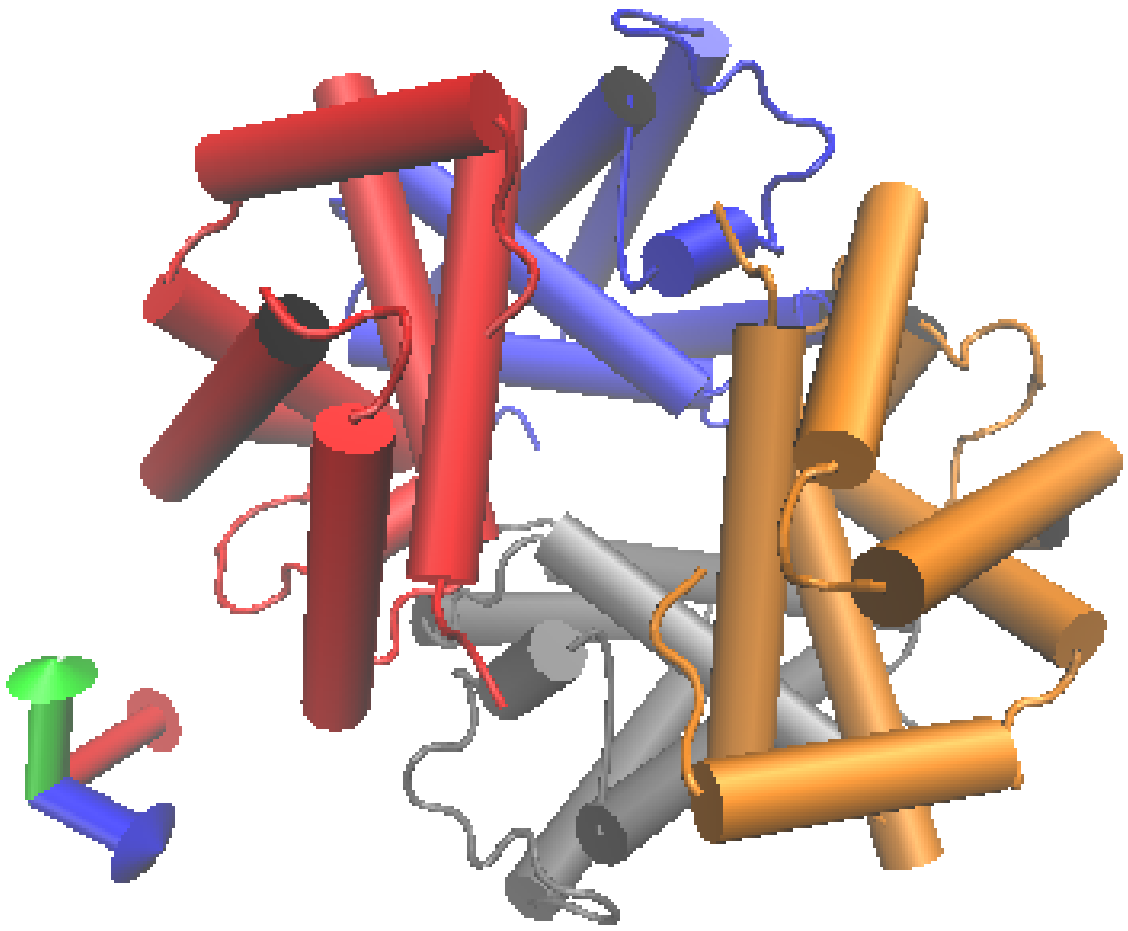


Figure 2.6 Proteins and their quaternary structures

This figure shows the quaternary structure of hemoglobin (1GZX.pdb) (Berman et al., 2000; Paoli et al., 1996) which has four protein chains, two α chains and two β chains, colored according to the residue type. It has a heme group, which binds oxygen atom(not shown). The structure was rendered using VMD (Humphrey et al., 1996).

CHAPTER 3. PROTEIN SECONDARY STRUCTURE PREDICTION USING KNOWLEDGE BASED POTENTIALS

A paper published in the Proceedings of IJCCI/ICNC 2010[1]

Saras Saraswathi [2,3] and Robert L. Jernigan[2], Andrzej Koliniski[4], Andrzej Kloczkowski[2,5]

Keywords

Protein secondary structure prediction, Neural networks, Extreme learning machine, Particle swarm optimization

Abstract

A novel method is proposed for predicting protein secondary structure using data derived from knowledge based potentials and Neural Networks. Potential energies for amino acid sequences in proteins are calculated using protein structures. An Extreme Learning Machine (**ELM**) classifier is used to model and predict protein secondary structures. Classifier performance is maximized using the Particle Swarm Optimization (**PSO**) algorithm. Preliminary results for **ELM-PSO** show improved results.

[1]Reprinted with permission of IJCCI/ICNC, 2010, ISBN 978-989-8425-32-4, pp. 370-375.

[2]Saras - Graduate student and Professors, respectively, Department of Biochemistry, Biophysics, and Molecular Biology, L .H. Baker Center for Bioinformatics and Biological Statistics, Iowa State University

[3]Primary researcher and author

[4]Professor, Laboratory of Theory of Biopolymers, Faculty of Chemistry, Warsaw University, Pasteura 1, 02-093 Warsaw

[5]Author for correspondence

3.1 Introduction

Large scale advances in genome sequencing and resultant availability of large numbers of proteins sequences has given protein secondary structure prediction increasing importance in computational biology. Improvements in secondary structure prediction can lead to progress in protein engineering and drug design. Existing crystallographic techniques are too expensive and time consuming for large-scale determination of protein three-dimensional structures. Prediction of secondary structures might be a useful intermediate step to speed up structure prediction (Lomize et al., 1999; Ortiz et al., 1999). Secondary structure prediction can assist in gene identification and classification of structures and functional motifs and in identifying malfunctioning structures which cause human diseases.

Several computational methods have been successfully used in secondary structure prediction, of which empirical and machine learning methods have proved to be the most successful (Chou and Fasman, 1974; Qian and Sejnowski, 1988; Ward et al., 2003). **GOR** was a pioneer method based on information theory (Garnier et al., 1978, 1996). Evolutionary information was used (Kloczkowski et al., 2002) for improved structure prediction. PredictProtein server (Rost et al., 2004) uses multiple sequence alignment based neural networks. The **PSIPRED** algorithm (Jones, 1999) uses **PSIBLAST** (Altschul et al., 1997) and neural networks. The **Jpred** prediction server (Cole et al., 2008) runs on the **Jnet** algorithm (Cuff and Barton, 2000). Large scale secondary structure prediction methods were developed (Montgomerie et al., 2006; Pollastri et al., 2007) using existing structural information and computational methods to claim an accuracy of 85.7% for sequences with over 30% sequence homology. It was suggested that long-range interactions are an important factor to be considered (Kihara, 2005) in order to achieve higher classification accuracy.

We propose a novel strategy for secondary structure prediction using knowledge based potential profiles. A two stage Extreme Learning Machine classifier called the **ELM-PSO**, is used for classification of secondary structures. **PSO** (Clerc and Kennedy, 2002) is used to improve the performance of the **ELM** classifier, by tuning its parameters such as input weights, bias and number of hidden neurons used in the neural network.

This paper is organized as follows: Section 2 gives a brief description of the data. Section 3 describes the two-stage **ELM-PSO** classification technique. Section 4 discusses the results and gives a comparative study followed by conclusions in Section 5.

3.2 Data generation using CABS force field

Data is derived based on **CABS** force-fields, (Kolinski, 2004), which includes information pertaining to long and short range interactions between amino acids in proteins. The dictionary of secondary structure assignment Database of Secondary Structure in Proteins (**DSSP**), (Kabsch and Sander, 1983), has 8 classes of protein secondary structures. We use only a reduced set of three secondary structures, namely, α-helix (**H**), β-strand (**E**) and coil (**C**). A profile matrix was created using **513** non-homologous (target) protein sequences from the **CB513** data set (Cuff and Barton, 2000), where the sequence homology is less than 30%. A comprehensive description of the data generation using the **CABS** algorithm is given under section 2.3 of this thesis.

3.3 Methods and optimization

An Extreme Learning Machine (**ELM**) (Huang et al., 2006) classifier, which is a form of Neural Network, is used for classification. **PSO** is used to tune the parameters of the ELM. The data was also evaluated using Support Vector Machine (SVM) and Naïve Bayes (**NB**) algorithms using the WEKA (Witten and Frank, 2005) software tool for classification.

3.3.1 Encoding of knowledge-based potential data

In a neural net framework, the input consists of a set of patterns (residues), each having a set of 27 features (profile values), which are normalized to values between 0 and 1. The output consists of three units which correspond to one of three secondary structure elements, represented as a 1 for the class of interest and a -1 for the other two classes. A given input is combined with a bias and a set of weights and is processed through an activation function at the hidden layer level. The output of the hidden layer is combined with another set of

weights to yield three outputs. The predicted class is considered as the output which has the maximum value, which corresponds to choosing the output with the smallest mean-squared error.

The knowledge based potential data used for classification is derived from the **CB513** (Cuff and Barton, 2000) set of protein sequences. The profile of each amino acid present in the protein sequences consists of 27 features for each of N amino acids, where N is the number of residues in a single protein. Of the 27 features, the first 9 features are the energy potentials related to α-helices (H), the next 9 features are related to β-strands (E) and the last 9 features are related to coils (C) as seen in Figure 3.1 and Figure 3.2 .

3.3.2 Scaling method used for secondary structure prediction

The relationship between the columns of the data and the three classes of secondary structures gives a particular advantage in getting better classification accuracy, since this information can be used as prior information during the training phase (although this information will not be available on a blind set or a new set of proteins). Based on this prior knowledge, class specific features of the target class can be given extra weights (importance) compared to the rest of the features that belong to the negative classes. Hence the class specific features of each class (9 columns per class) were scaled (values boosted) according to a predetermined factor prior to building a training model. These factors (not unique) were obtained by brute force trial and error method, where selection was based on getting better classification results. It is noteworthy that the classification accuracy after this scaling depends on the scaling factors used, and ranges from 60% (for non-scaled data or data scaled with sub-optimal boosting values), to over 95%, when the optimal scaling factors are used. The first 9 features of all samples belonging to the H class, were scaled by a factor of 5, while the second set of 9 features were scaled by a factor of 3 and the last set of 9 features were scaled by a factor of 8. The scaling of data improves the classification accuracy considerably during the training phase. Samples which were scaled according to their classes were used for the 10-fold cross-validation in WEKA (Witten and Frank, 2005), which gave very high results for **SVM**

and Naïve Bayes algorithms. Since it is not possible to perform class-specific feature scaling during testing (blind) phase for the **ELM** method, three sets of test samples were generated for each sample in the test set. The first set had the first 9 features boosted in the same ratio as for the H class for all samples. The second set of test samples had the next set of 9 features boosted according to the factor used for the E class for all samples and the third set of test samples had the last set of 9 features scaled according to the factor used for the C class for all samples. Each test set was sent in turn and the votes were collected for the classification. For robustness, ten sets of training models were used to get the classification results for the same test set. Each training model yielded a set of three votes for each sample. These votes were all gathered to determine the class which receives the maximum number of votes. The results for the classification accuracies with and without feature scaling (value boosting) are given in the results section. Blind testing with voting was not done for **SVM** and Naïve Bayes algorithms since it would require modification of WEKA code.

3.3.3 Two-stage Extreme Learning Machine

The **ELM-PSO** consists of the Extreme Learning Machine (**ELM**) classifier (Huang et al., 2006) as the main algorithm, which uses a set of training samples to build a model. During the training phase, **PSO** is called upon to optimize the parameters, such as weights, number of hidden neurons and bias of the ELM, which results in improved classification accuracy. These parameters are stored and used during the testing phase. **ELM** is an improved version of a feed-forward neural network consisting of a single hidden layer. The initial set of input weights are chosen randomly, but they are tuned later by the PSO. The output weights from the hidden layer to the output layer are analytically calculated, using a pseudo inverse. A sigmoidal activation function is used for the hidden layer and a linear activation function is used for the output neurons. A comprehensive description of the **ELM** algorithm is given under Section 1.4.1 of this thesis. The simple steps involved in the **ELM** algorithm are:

- Given training samples and class labels (X_i, Y_i), select the appropriate activation function $G(.)$ and number of hidden neurons;

- Randomly select the input weights (V), bias (b) and calculate the output weights W analytically where $W = YY_h^{\dagger}$.
- Use the calculated weights (W, V, b) for estimating the class label in the test set and try to minimize the error between the observed and predicted values. Generalization performance depends on the choice of these parameters. They are later tuned by the **PSO** algorithm.
- The class label is estimated as the maximum value of K outputs y_i^k.

$$\hat{c_i} = \arg \max_{k=1,2,...,C} y_i^k. \tag{3.1}$$

It has been shown (Suresh et al., 2010) that the random selection of input weights (V) and bias (b) affects the generalization performance of the **ELM** multiclass classifier significantly resulting in large variances in testing accuracies. It has been shown (Saraswathi et al., 2011) that proper selection of **ELM** parameters (input weights, bias values, and hidden neurons) influences the generalization performance of the **ELM** multiclass classifier favorably by minimizing the error defined as:

$$\{H^*, V^*, b^*\} = \arg \min_{H,V,b} \{Y - T\} \tag{3.2}$$

where Y is the observed class value and T is the calculated output value of the class, for a given set of hidden neurons H and input parameters V and b. The best weights and bias values (denoted with the * symbols near the parameters) for the **ELM** can be found using search techniques and optimization methods that are not very computationally intensive. In this study, we use Particle Swarm Optimization for tuning the **ELM** parameters (H, V, b).

3.3.4 Particle Swarm Optimization

Particle Swarm Optimization (**PSO**) is a stochastic optimization technique (Kennedy and Eberhart, 1995; Clerc and Kennedy, 2002). This method mimics the intelligent social behavior

of flocks of birds or schools of fish, represented as particles in a population. These particles work together to find a simple and optimal solution to a problem in the shortest possible time. The **PSO** algorithm is initialized with a set of random values called particles which contribute collectively to the desired solution. These values represent various parameters that we hope to tune in order to improve performance. The algorithm iteratively searches a multi-dimensional space for the best possible solution, determined by a fitness criterion. **PSO** will find the best combination of hidden neurons, input weights, and bias values and return the (training) validation efficiency obtained by the **ELM** algorithm along with the best **ELM** parameters to obtain better generalization performance. The best parameters are stored and used during the testing phase. A comprehensive description of the **PSO** algorithm is given under Section 1.4.2 of this thesis.

3.4 Results and discussion

Several training models were built using **ELM** and two other algorithms, namely **SVM** and Naïve Bayes (**NB**) from the WEKA (Witten and Frank, 2005) suit of software for data classification. A 10-fold cross validation was performed for **SVM** and **NB** , where 90% of the proteins were used to build the training model while the remaining 10% were retained for testing the model, but all input information was scaled according to previously described values. A blind test was conducted using **ELM** with 4797 proteins for training and 4835 for testing. These residues were selected from a random selection of 30 proteins for the training set out of 400 proteins, while the test samples came from a separate set of 41 proteins retained for testing. Preliminary studies for the **ELM-PSO** classifier, **SVM** and **NB** show high accuracies of around 99% for the scaled training as seen in Table 3.2, while the results for the un-scaled version of the data, as seen in Table 3.1, is much lower at only 60% or less. The un-scaled version of the data uses only row specific feature information while the scaled data also uses column specific class information which increases the accuracy considerably. The lower testing accuracy of 94.4% for the **ELM** (blind) tested 4835 samples might be due to the smaller number of residues tested as compared to the other two models built from **SVM** and

Table 3.1 Confusion matrix and accuracies without feature scaling.

This table gives the results for the three classes of secondary structures, using **ELM-PSO**, **SVM** and Naïve Bayes, using data without feature scaling.

Confusion Matrix - ELM-PSO - Without feature scaling					
	H	**E**	**C**	**% Correct**	**Category**
H	1147	116	457	66.7	Q_H
E	300	329	474	27.1	Q_E
C	604	175	1195	30.6	Q_C
				55.7	Q_3
Confusion Matrix - SVM - Without feature scaling					
	H	**E**	**C**	**% Correct**	**Category**
H	3153	533	1672	58.8	Q_H
E	817	1353	1411	22.8	Q_E
C	1446	595	5083	20.5	Q_C
				59.7	Q_3
				58.5	F-Measure
				70.0	**AUC**
Confusion Matrix - Naïve Bayes - Without feature scaling					
	H	**E**	**C**	**% Correct**	**Category**
H	3244	1217	897	60.2	Q_H
E	705	2168	708	60.1	Q_E
C	2028	2168	708	47.6	Q_C
				54.8	Q_3
				55.1	F-Measure
				73.5	**AUC**

NB with the full data set. The **ELM** classifier trains on sets of 2000 to 3000 samples at a time and builds several of these models by selecting samples at random from the pool of available training samples (from the 400 training proteins), a very computationally intensive process. The parameters for every **ELM** model are optimized by calling **PSO** and a single pattern from the test set is repeatedly tested by each model, giving a consensus classification for the type of the test sample. The class that occurs with the highest frequency in these classifications is taken to be the predicted class for this test sample. Preliminary results for a set of 4835 test samples are given in Table 3.1 and Table 3.2 for scaled and un-scaled data. On the other hand the high accuracies for **SVM** and **NB** can be attributed to the technique of cross vali-

Table 3.2 Confusion matrix and accuracies with feature scaling.

This table gives the results for the three classes of secondary structures, for data *with feature scaling*, using **ELM-PSO**, **SVM** and Naïve Bayes algorithms.

Confusion Matrix - ELM-PSO - With feature scaling					
	H	**E**	**C**	**% Correct**	Category
H	1814	0	0	100.0	Q_H
E	56	942	0	94.3	Q_E
C	224	0	1799	89.9	Q_C
				94.4	Q_3
Confusion Matrix - SVM - With feature scaling					
	H	**E**	**C**	**% Correct**	Category
H	24854	67	8	99.7	Q_H
E	0	16879	4	100.0	Q_E
C	0	0	31096	100.0	Q_C
				99.9	Q_3
				99.8	F-Measure
				99.9	**AUC**
Confusion Matrix - Naïve Bayes - With feature scaling					
	H	**E**	**C**	**% Correct**	Category
H	24896	33	0	99.9	Q_H
E	256	16627	0	98.5	Q_E
C	0	19	31077	99.9	Q_C
				99.6	Q_3
				99.6	F-Measure
				100.0	**AUC**

dation where the input data is uniformly scaled according to previous criteria, using feature specific class information, which results in higher accuracy. There is no blind test of data. So, unless the algorithm can discern this feature specific pattern automatically without involving the computationally intensive brute force method for finding the scaling parameters that was used here, it is not very practical. Future work will aim to improve the **ELM-PSO** algorithm to learn this prior information automatically. Table 3.3 shows that the **ELM-PSO** methods perform very well compared to other studies in the literature for scaled data. The accuracy on the un-scaled data is lower for all models and is comparatively low for the blind test, indicating that the learning algorithm needs further tuning to discern the column-wise

information during (blind) testing phase. The column-wise class information is a unique feature of our data that separates the three classes linearly and hence gives high results. Table 3.1 and Table 3.2 also give the **F-measure** and area under the curve (**AUC**) values for **SVM** and Naïve Bayes classifications. These calculations help us to gauge the quality of the predictions.

The performance of classifications can be evaluated in terms of the true positives (**TP**-correct) and false positive (**FP**-error) terms. Similar definition holds for true negatives (**TN**) and false negatives (**FN**). The output of a classification might provide estimated probabilities which determine the predicted class according to a pre-set threshold. **TP** rate and **FP** rate can be graphed as coordinate pairs which form the receiver operating characteristic curve (ROC curve). The area under the **ROC** curve (**AUC** or **AUROC**) helps to aggregate the performance of all the testing results, where a higher value closer to 1.00 denotes perfect performance. **F-measure** gives the test's accuracy. It uses *precision p* and *recall r* of the test, where *p* is the ratio of correct results divided by *all returned results (***TP***/(***TP***+***FP***))* and *r* is the number of correct results divided by the number of *expected results (***TP***/(***TP***+***FN***))*. **F-measure** is calculated as given in Equation 3.3, where the best score for **F-measure** can be as high as 1 and the worst score can be as low as 0.

$$\mathbf{F-measure} = 2 * (precision * recall)/(precision + recall) \tag{3.3}$$

Table 3.3 Comparison study of results for secondary structure prediction

This table compares the results of **ELM-PSO** based prediction results, on data with scaled features, with other studies in literature.

Method	$\mathbf{Q_3\%}$	$\mathbf{Q_H\%}$	$\mathbf{Q_E\%}$	$\mathbf{Q_C\%}$
PHDRost and Sander (1993)	70.8	72.2	66.0	72.0
JNet server (Cuff and Barton, 2000)	76.4	78.4	63.9	80.6
SVMpsi (Kim and Park, 2003)	76.6	78.1	65.6	81.1
SPINE server Dor and Zhou (2007)	80.0	84.4	72.2	80.5
ELMPSO with feature scaling (our study)	**94.4**	**100.0**	**94.3**	**89.9**

3.5 Conclusions

A two stage approach for secondary structure prediction was presented where an Extreme Learning Machine (neural network) was used along with Particle Swarm Optimization (**ELM-PSO**) for classifying a reduced set of three secondary structures, namely, α-helix, β-strand and coil. The data was generated using **CABS** potential energy. **ELM-PSO** needs improvement to achieve better accuracies on blind tests so that comparative results can be achieved on new proteins.

Acknowledgements

We thank Pawel Gniewk, a student at *Theory of Biopolymers, Faculty of Chemistry, Warsaw University*, whose original idea and algorithm was used to generate the potentials data that was used for the secondary structure predictions. We acknowledge the support of National Institutes of Health through grants R01GM081680, R01GM072014, and R01GM073095 and the support of the NSF grant through IGERT-0504304.

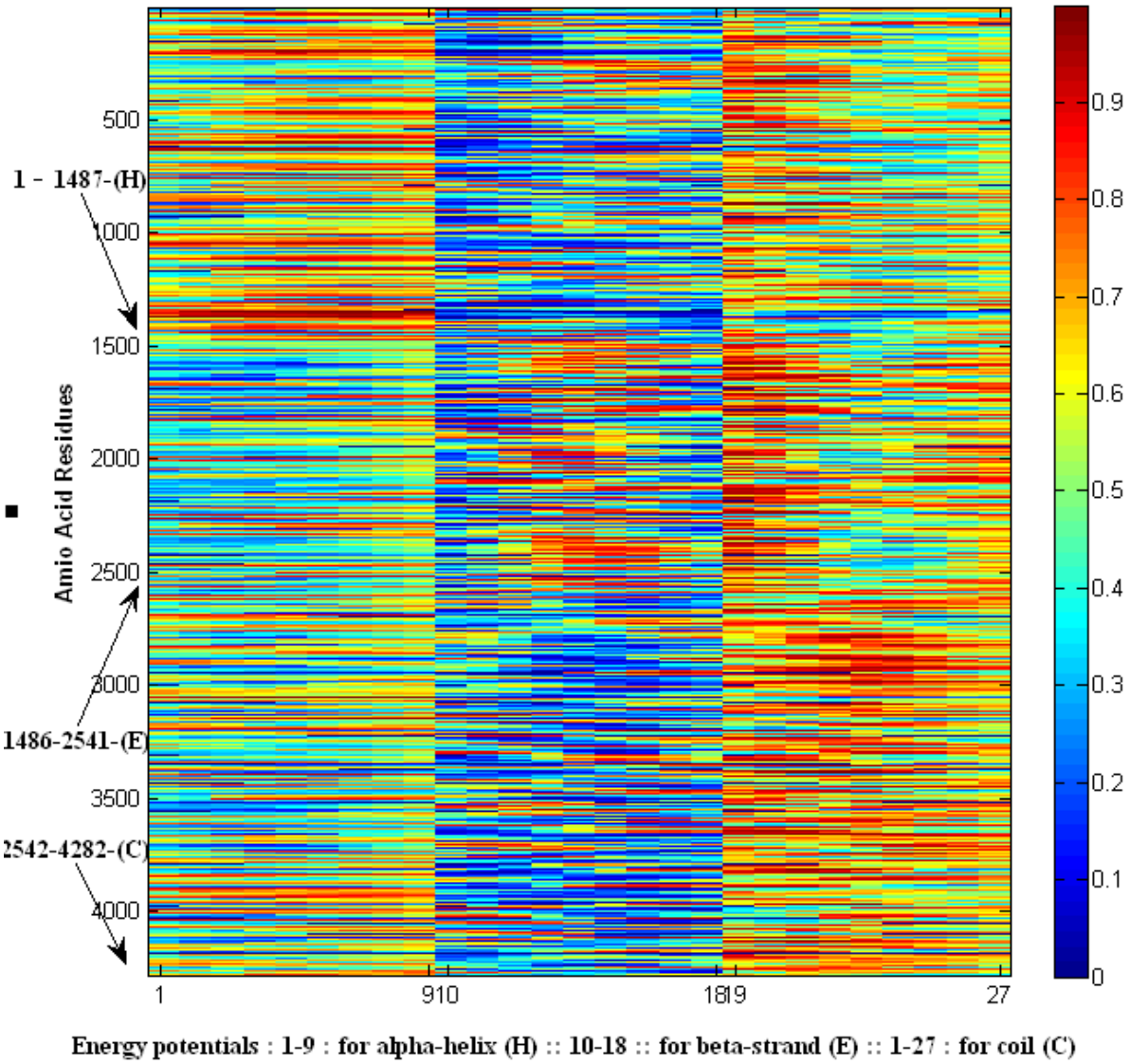


Figure 3.1 Visualization of data without feature scaling.

Energy potentials are represented along the x-axis, the first 9 features belong to helix (H), the next 9 features are for strand (E) and the last set of features 19 - 27 for coil (C). The color intensity indicates the value of the potential energy, with a dark blue for a low value and a red indicates a high value. The residues (total: 4282) along the y-axis have been sorted according to the three classes, where residues 1 - 1487 belong to class H, 1488 - 2541 belong to class E and 2542 - 4282 belong to class C. Note: there is not much horizontal differentiation among the three classes which becomes evident in Figure 3.2, after data is subjected to feature specific scaling. Results for classification of this *un-scaled* data is given in Table 3.1. These results are discussed further in the results section.

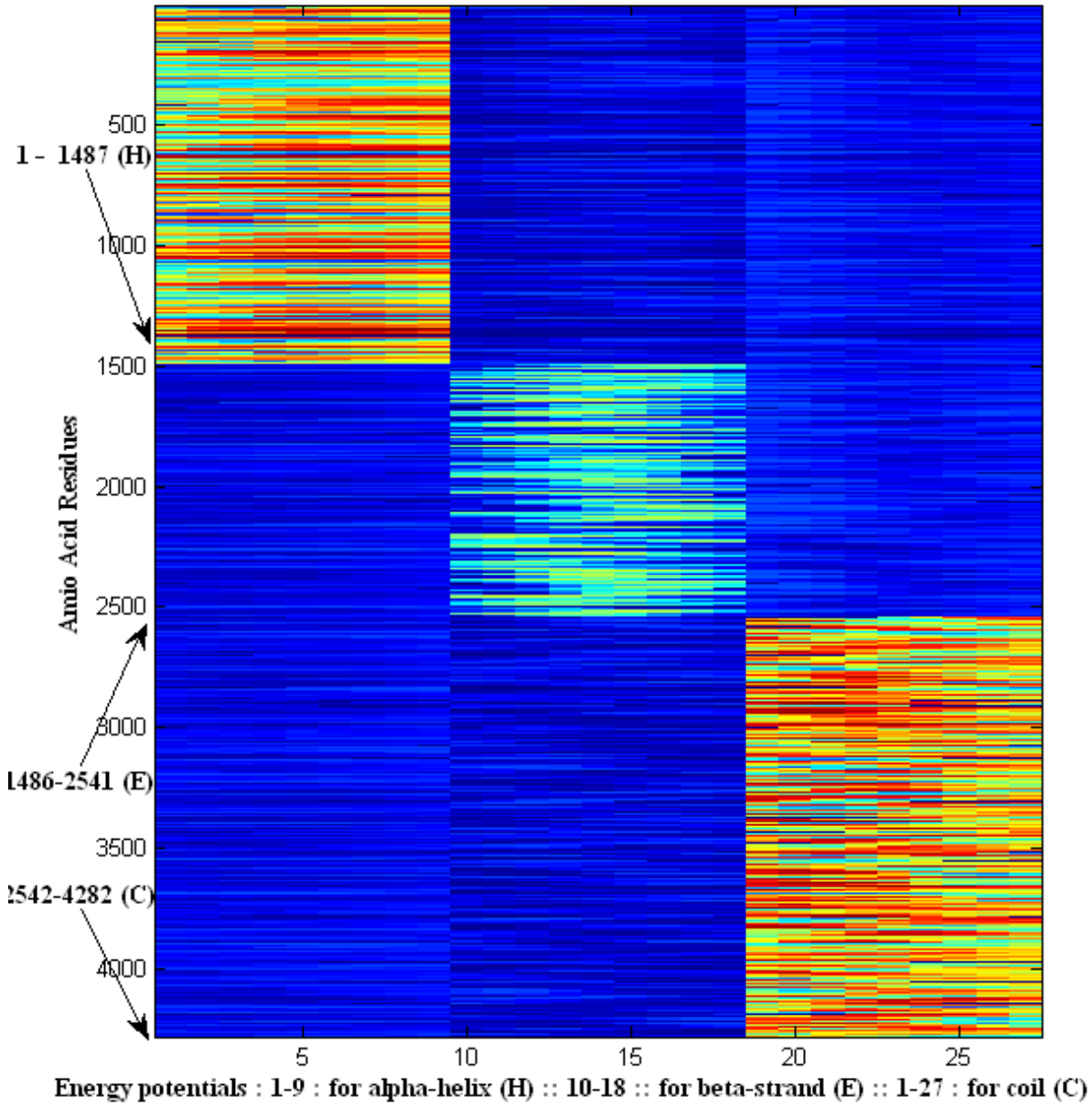


Figure 3.2 Visualization of data with feature scaling.

The same sample data shown in Figure 3.1, is given here *after feature scaling*. Descriptions of the X,Y axes and colors are the same as given in Figure 3.1. Compared to Figure 3.1, it can be seen that class-specific feature scaling provides for a distinct separation of the classes, which results in higher accuracy during classification, using ELM, SVM-SMO and Naïve Bayes algorithms, with results shown in Table 3.2. These results are discussed further in the results section.

REFERENCES

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST a new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402.

Chou, P. Y. and Fasman, G. D. (1974). Prediction of protein conformation. *Biochemistry*, 13:222–245.

Clerc, M. and Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans Evolutionary Comput*, 6:58–73.

Cole, C., Barber, J. D., and Barton, G. J. (2008). The Jpred 3 secondary structure prediction server. *Nucleic Acids Research, Web Server issue*, 36:W197–W201.

Cuff, J. A. and Barton, G. J. (2000). Application of multiple sequence alignment profiles to improve protein secondary structure prediction. *Proteins*, 40:502–511.

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

Garnier, J., Gibrat, J. F., and Robson, B. (1996). GOR secondary structure prediction method version IV. *Methods in Enzymology*, 226:540–553.

Garnier, J., Osguthorpe, D. J., and Robson, B. (1978). Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *Journal of Molecular Biology*, 1:97–120.

Huang, G. B., Zhu, Q. Y., and K, S. C. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, 70:489–501.

Jones, D. (1999). Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology*, 292:195–202.

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

Kennedy, J. and Eberhart, R. C. (1995). Particle Swarm Optimization. *Proceedings of the IEEE International Conference on Neural Networks*, 4:1942–1948.

Kihara, D. (2005). The effect of long-range interactions on the secondary structure formation of proteins. *Protein Science*, 14:1955–1963.

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

Kloczkowski, A., Ting, K. L., Jernigan, R. L., and Garnier, J. (2002). Combining the GOR V algorithm with evolutionary information for protein secondary structure prediction from amino acid sequence. *Proteins*, 49:154–166.

Kolinski, A. (2004). Protein modeling and structure prediction with a reduced representation. *Acta Biochem Pol.*, 51:349–371.

Lomize, A. L., Pogozheva, I. D., and Mosberg, H. I. (1999). Prediction of protein structure : The problem of fold multiplicity. *Proteins*, 37:199–203.

Montgomerie, S., Sundaraj, S., Gallin, W., , and Wishart, D. (2006). Improving the accuracy of protein secondary structure prediction using structural alignment. *BMC Bioinformatics*, 301:301.

Ortiz, A. R., Kolinski, A., Rotkiewicz, P., Ilkowski, B., and Skolnick, J. (1999). Ab initio folding of proteins using restraints derived from evolutionary information. *Proteins*, 37:177–185.

Pollastri, G., Martin, A., Mooney, C., and Vullo, A. (2007). Accurate prediction of protein secondary structure and solvent accessibility by consensus combiners of sequence and structure information. *BMC Bioinformatics*, 8:201.

Qian, N. and Sejnowski, T. J. (1988). Predicting the secondary structure of globular proteins using neural network models. *Journal of Molecular Biology*, 202:865–884.

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

Rost, B., Yachdav, G., and Liu, J. (2004). The predictprotein server. *Nucleic Acids Research*, 32:W321–W326.

Saraswathi, S., Suresh, S., and Sundararajan, N. (2011). Icga-pso-elm approach for accurate multiclass cancer classification resulting in reduced gene sets in which genes encoding secreted proteins are highly represented. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 8:452–463.

Suresh, S., Saraswathi, S., and Sundararajan, N. (2010). Performance enhancement of extreme learning machine for multi-category sparse cancer classification. *Engineering Applications of Artificial Intelligence*, 23:1149–1157.

Ward, J. J., McGuffin, L. J., Buxton, B. F., and Jones, D. T. (2003). Secondary structure prediction with support vector machines. *Bioinformatics*, 19:1650–1655.

Witten, I. H. and Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco.

CHAPTER 4. FLOPRED FOR SECONDARY STRUCTURE PREDICTION USING KNOWLEDGE-BASED POTENTIALS

[1]

4.1 Introduction

Previous studies applying the **PSO-ELM** algorithm to a set of proteins from the **CB513** dataset gave high prediction results when prior information was included as part of the input information, as discussed in Section 3.3.2 on page 70. The knowledge-based potential data consists of 27 features, where the first 9 features belong to α-helix, the next 9 features belong to β-sheet and the last 9 features belong to coil. This prior information was used in initial studies to investigate the nature of the encoded data but not in later studies. Our aim, in this study is to develop an algorithm that learns from the information encoded in protein sequences from given feature sets with no user intervention.

4.2 FLOPRED Methodology for secondary structure prediction

The **FLOPRED** algorithm was developed using **ELM** as explained in Section 1.4.1 on page 6 and advanced **PSO** algorithms as explained in Section 1.4.2 on page 11. By using **FLOPRED**, we achieve accuracies between 86% and 92%, which are better than previously reported in the literature for similar studies.

[1] In preparation for submission to journal

4.3 Data generation

The data for this study was generated and encoded using the **CB513** data set (Cuff and Barton, 2000) and the CABS algorithm (Kolinski, 2004) as discussed in Section 2.3 on page 46 and Section 3.3.1 on page 69. Several training models were built using **ELM** as the classifier which calls on **PSO** for optimization. Two sets of data were built to test the **FLOPRED** algorithm as described below. Part of each dataset was selected for training models and the remaining independent sets of sequences were tested for classification accuracy.

- **Dataset-**84: The first set of data consists of a selection of a small set of 84 proteins, with 7,500 residues, hitherto referred to as dataset-84. The selection criteria for this set was the requirement that the sequence length had to be less than 125 residues, to keep computational time and resources within manageable limits for the initial study, due to the complex nature of the CABS algorithm for data generation.

- **Dataset-**415: A larger set of 415 proteins with 62,000 residues (hitherto referred to as dataset-415) were selected from the **CB513** data set, which includes the initial set of 84 proteins. The criteria for sequence selection in this data was to select sequences which did not share a pair-wise homology of greater than 70% with any of the templates. The overall homology distribution between the template and the target sequences can be seen to be between 10% and 18% as shown in the histogram in Figure 2.1. The remaining 113 of the 513 proteins were discarded from this study since they were found to be either homologous to the template sequences as discussed in Section 2.3.7 on page 50 or they were homologous to the template structures as discussed in section 2.3.8 on page 51. The list of templates and sequences used in each of these studies are given in Appendix A,B and C on page 178.

  The results of studies conducted using these two datasets are discussed below in the results section.

4.3.1 Parameters used for PSO

Table 4.1 gives the parameters used by PSO and ELM algorithms. Max-iteration is the number of iterations (50) the PSO goes through to select the best parameters while swarm-size (100) is the number of particles used during each search. Lambda is one of the ELM parameters used in the sigmoidal activation function. The search space ($27 * 1,250$) is needed to represent the weights of the training network where 27 is the number of features and $1,250$ is the number of hidden neurons. Another $1,250$ values are needed for the bias of the hidden layer, for a total of $37,350 + 1,250 = 38,600$ parameters in the search space. Although this seems to be a large number, we are using thousands of samples to train the model and this number is still much less than what would be needed for over fitting the model. We need 100 sets of these parameters in each of the 50 iterations, since there are 100 particles used in each search. In each of the 50 iterations 100 sets of $38,600$ values are randomly generated and each set is evaluated by ELM. The particle which holds the set of $38,600$ values closest to the desired result (the parameters which give the minimum error to give the best classification accuracy) is considered as the best set of parameters. The best set from each of the 50 iterations is stored and the overall best set from all the 50 iterations is finally used during the testing of the independent set of data.

The parameters for the maximum number of iterations, swarm size, lambda and the number of hidden neurons can vary widely if selected manually by the user. Higher values of these parameters can mean there has been over-fitting of the training data, whereas lower values mean under-fitting. Either of these two extremes will result in lower generalization performance resulting in lower accuracies. **FLOPRED** is able to select the minimum number of each of these parameters needed to achieve high testing accuracies while maintaining lower standard deviations, resulting in excellent generalization performance. Each of these parameters have been included for optimization by the **PSO** and validated by **FLOPRED** at different stages of our research, resulting in improved accuracies from an initial $\mathbf{Q_3}$ testing accuracy of 79% to the final results

obtained for the two data sets exceeding 92%.

4.4 Results and discussion

Table 4.1 Parameters used for PSO and ELM

This table gives the parameters used by PSO and ELM algorithms. Max-iteration is the number of iterations the PSO goes through to select the best parameters. Swarm-size (100) is the number of particles in each iteration. The search space is $(27 * 1,250)$ for the weights of the training network where 27 is the number of features and $1,250$ is the number of hidden neurons. Another $1,250$ values are needed for the bias values, for a total of $37,350 + 1,250 = 38,600$ parameters in the search space. Lambda is one of the ELM parameters used in the sigmoidal activation function. In each of the 50 iterations 100 sets of $38,600$ randomly generated values are evaluated by ELM. Finally, the overall best set of $38,600$ values are stored for use during the testing of the independent testing data. These parameters are discussed further under Section 4.3.1 on page 85.

Parameters	**Values**
Max-iteration	50
Swarm-size	100
ELM-Features	27
ELM-Hidden Neurons	1,250
ELM-lamda	0.016

4.4.1 Results for dataset-84: Performance metrics

We use dataset-84, where $4,000$ residues are used to build a training model and the remaining $2,647$ residues are used for secondary structure prediction. The training set has a good mix of the three secondary classes, α-helix,β-strand and coil, represented in the proportion in which they naturally occur in the sequences. For this dataset, we are able to predict secondary structures with a higher $\mathbf{Q_3}$ training accuracy of 93.33% and a testing accuracy of 92.24% with a standard deviation of 0.48%, on a small group of 84 proteins, which shows a good generalization performance. The contribution to these high accuracies come from the high individual testing accuracies of 94.19% for α-helix, 92.39% for β-strand and 91.11% for coil. We observe a Matthew's correlation-coefficient ranging between 80.58% and 84.30% for

the three secondary structure classes. For dataset-415 we obtain a testing $\mathbf{Q_3}$ accuracy of 85.16% with improved performance. These are the most recent results.

Table 4.2 Metrics of the testing results for dataset-84

This table gives various metrics for the results of classification using dataset-84, averaged over 25 separate runs. 4,000 residues were used for the training model and 2,647 residues were used for testing. The same training and testing set was used in all runs. All metrics given in the table, including $\mathbf{Q_3}$ and **MCC** accuracies are defined and explained in Section 2.1.7 on page 35. These results are illustrated in Figure 4.1 and are discussed further under Section 4.4.1

Metrics	α**-helix**	β**-sheet**	**Coil**	**Average**	**Stdev**
$\mathbf{Q_3}$-training	94.42	88.05	90.18	**90.89**	0.33
$\mathbf{Q_3}$-testing	88.41	87.53	93.15	**89.70**	0.48
Sensitivity (recall)	70.50	80.42	87.64	79.52	0.50
Specificity	97.67	93.93	90.57	94.06	0.23
Matthew's-corr-coeff (MCC)	0.74	0.76	0.78	0.76	0.61
+ve Predictive Value (precision)	93.15	87.53	88.41	89.70	0.48
+ve Predictive Value-Prev	92.35	**83.56** *	87.85	**87.92**	0.50
-ve Predictive Value	88.05	90.05	89.92	89.34	0.35
-ve Predictive Value-Prev	89.26	**92.59** *	90.39	**90.75**	0.23
False +ve Rate (Type I error)	2.33	6.07	9.43	5.94	0.23
False -ve Rate (Type II error)	29.50	19.58	12.36	20.48	0.50
Likelihood Ratio +ve	30.39	13.28	9.30	17.65	0.96
Likelihood Ratio -ve	0.30	0.20	0.13	0.21	0.08

The results of a previous study, with slightly lower accuracies, are given in Table 4.2 on page 87. In this study 25 different sets of data are created. The same training and testing set is used on all runs and the individual results are averaged over these 25 runs. Various training and testing accuracies are given in Table 4.2 and Figure 4.1. This figure shows accuracies for the three classes of secondary structures α-helix, β-strand and coil and the average $\mathbf{Q_3}$ accuracies. $\mathbf{Q_3}$-training, $\mathbf{Q_3}$-testing, sensitivity, specificity, Matthew's Correlation Coefficient (**MCC**), (Positive Predictive value (**PPV**), Negative Predictive Value (**NPV**) with and without

Table 4.3 Post-test probabilities for dataset-84

This table compares the post-test probabilities for positive and negative classifications of the three secondary structures. Sensitivity (Sen) and Specificity (Spc) are used to calculate the Likelihood Ratio Positive (LR+ve) and Negative (LR-ve) values. The Prevalence (Prev) or common occurrence of secondary structure percentage in the dataset and Pre-test odds (P-odds) is also taken into consideration as post-test probabilities. The post-test probabilities are calculated when the classification is given as positive (P+ve) and when it is given as negative (P-ve). The **D+ve** gives the difference between pre-test and post-test probabilities when classification is positive. The **D-ve** gives the difference between pre-test and post-test probabilities when a classification is negative. We see that the post-test probability percentages for positive classifications are better since there is a greater difference for them compared to the post-test probabilities when the classifications are negative. This shows that this model yields better confidence in classifying positive cases than ruling out negative cases. The calculations for these results are discussed in Section 4.4.1

	Sen	**Spc**	**Prev**	**LR+ve**	**LR-ve**	**P-odds**	**P+ve%**	**D+ve**	**P-ve%**	**D-ve**
α-helix	0.71	0.98	0.36	30.26	0.30	1.81	0.94	**0.59**	0.14	0.21
β-sheet	0.80	0.94	0.22	13.25	0.21	3.45	0.79	**0.57**	0.06	0.17
Coil	0.88	0.91	0.42	9.29	0.14	1.39	0.87	**0.45**	0.09	0.33

prevalence), are given. All metrics given in the table are defined and explained in Section 2.1.7 on page 35.

Here we see an average $\mathbf{Q_3}$ training accuracy of 90.89% while the individual training accuracies are 94.42%, 88.05% and 90.18% respectively for the three secondary structures with a standard deviation of 0.33% for training and 0.48% for testing which are small compared to other reported values. The small standard deviations and the small difference of less than 1.19% between the training and testing accuracies show good generalization performance indicating that we have built a good model. We see an average $\mathbf{Q_3}$ testing accuracy of 89.70% while the individual testing accuracies are 88.41%, 87.53% and 93.15% respectively for the three classes. Similarly, the other metrics seen in this table such as the average correlation coefficients 0.76, specificity 94.06, sensitivity 79.52 and precision 89.70 are also higher compared to those found in the literature (not shown in the comparison table; when the final predictions are higher then, all other metrics will also show correspondingly higher values). We have val-

ues for **MCC** ranging between 0.74 and 0.78, which shows that the observed and predicted values are well correlated indicating good prediction results. Our MCC is higher than seen for other results in the literature as shown in Table 4.5.

The positive predictive value for β-sheet is much lower when prevalence is taken into consideration, indicating that beta-sheets may not be sufficiently represented in the training model. There is a 4% difference between the **PPV** and **PPV** with prevalence. In general all the three secondary structures show lower accuracies when prevalence of the structures is taken into consideration, showing that $\mathbf{Q_3}$ accuracies give an optimistic view of the predictions when prevalence is not taken into consideration. There is a 2.22% reduction in the final positive predictive value with prevalence. All the values for **NPV** show higher values with prevalence although the gain is less than 1% for α-helix and coil while the biggest gain is for β-sheet at 2.54%, with a final overall gain of 1.41% with prevalence. Initially, the **PPV** and the **NPV** values seem to have values close to each other, with only a difference of 0.44% between them, indicating that this model might have the same level of discrimination between positive and negative classes. But, after we take prevalence into consideration, we can see that the model is much better at classifying negative classes than classifying positive classes. There is a difference of 2.77% between these values when we take prevalence into account. This type of analysis can help us to improve the model and modify it to obtain better positive predictive values.

On comparing the false positive (**FPR**) and false negative (**FNR**) rates, we see that there are four times more false negatives at 20.48% than false positives at 5.94%. The worst **FNR** seems to be for α-helix followed by β-sheet, while the accuracies seem to reverse themselves in quality for **FPR**. This information can be used to improve the model to reduce its **FNR**, which will help to increase the accuracies of classification.

Next, we look at post-test probabilities for these tests. The Likelihood Ratio +ve **LRP** show large values for all three secondary structures. As discussed in the definition for these metrics, if the **LRP** values are greater than 8 then there is a large increase in the odds that these evaluations indeed belong to the positive classes as classified by the algorithm. The

LRN do not show large gains. Post-test probabilities are analyzed further and the results are shown in Table 4.3 on page 88. In this table we see that the post-test probability percentages for positive classifications are better, with an average of 52% increase compared to only 9% increase for negative classes. This shows that this model has better post-test confidence in classifying positive cases than ruling out negative cases. This information could be used for fine-tuning the **SSP** accuracies when combined with other criteria such as propensities of certain amino acids to appear at the ends of secondary structures or other physicochemical properties of amino acids. Although the **PPV-Prev** values indicate that the model might be good for predicting negative classes, the post-test probabilities show high confidence in predicting positive classes.

It is to be noted that such high accuracies are not seen in the literature for any methods as shown in Table 4.5. The high accuracies can be attributed to the advanced PSO algorithms Fernández-Martínez and García-Gonzalo (2010) used. Initially when these algorithms are used to tune only the weights and biases of ELM they provided accuracies which were about 5% higher than simpler PSO algorithms. Later on these algorithms are optimized to include more ELM parameters like the number of hidden neurons, lambda and other PSO parameters specific to the advanced algorithms, and this helped to tune the ELM further to provide very high accuracies as seen in these results.

4.4.2 Results for dataset-415: Performance metrics

Dataset-415 underwent a 5-fold cross-validation test, using the sequences in dataset-415. The ELM and PSO parameters used are similar to the ones used for dataset-84 and the searching techniques are the same as discussed above. The dataset of 415 proteins is divided into five equal numbers of proteins. Each of the five sets is retained for testing, iteratively, while the remaining four sets are used for building training models. All the four training sets are concatenated into one large training set and five models are built, each with approximately 6647 to 9000 residues. The training set is divided into smaller subsets (models) to facilitate computation. The ELM algorithm uses a pseudo-inverse to analytically calculate some of its

Table 4.4 Metrics of the testing results for dataset-415

This table gives various metrics for the results of classification using dataset-415, with a 5-fold cross-validation test. **MCC** is the Matthew's correlation-coefficient. These results are discussed further under Section 4.4.2

Metrics	**α-helix**	**β-sheet**	**Coil**	**Average**	**Stdev**
Q_3-training	91.78	78.83	82.99	84.53	0.50
Q_3-testing	79.08	75.58	90.37	81.67	1.38
Sensitivity (recall)	67.35	60.06	78.62	68.67	2.38
Specificity	94.09	91.60	84.65	90.12	0.81
Matthew's-corr-coeff (MCC)	0.65	0.56	0.63	0.61	1.91
+ve Predictive Value (precision)	90.37	75.58	79.08	81.67	1.38
+ve Predictive Value-Prev	85.16	69.30	79.16	77.88	1.17
-ve Predictive Value	77.82	84.14	84.26	82.07	1.23
-ve Predictive Value-Prev	85.16	87.92	84.22	85.77	1.06
False +ve Rate (Type I error)	5.91	8.40	15.35	9.88	0.81
False -ve Rate (Type II error)	32.65	39.94	21.38	31.33	2.38
Likelihood Ratio +ve	11.49	7.18	5.14	7.94	0.64
Likelihood Ratio -ve	2.90	2.30	3.97	3.06	0.22

parameters and it is difficult and computationally intensive to invert large matrices. Since each protein has a different number of residues, the data sets differ in the number of residues and the composition of amino acids contained in them. But since whole proteins are selected randomly from the initial set to make five groups, the datasets has a reasonably similar distribution of amino acids as seen in normal proteins. There are between $10,000$ and $12,000$ residues in each test set. Best parameters selected in each of the 5 models are tested using the *same* set of sequences (one of five initial sets set up for cross-validation) that are set aside for testing purposes during each cross-validation. This results in five votes (classification), for each residue in the testing set, one from each model. The class with the highest number of votes gathered from these models is taken as the predicted secondary structure for a given residue in the test set. This exercise yields an average cross-validation training accuracy of

84.53% with a standard deviation of 0.5% and a testing accuracy of 81.67% with a standard deviation of 1.38% as given in Table 4.4.

Here we see an average Q_3 training accuracy of 84.53% while the individual training accuracies are 91.78%, 78.83% and 82.99% respectively for the three secondary structures with a standard deviation of 0.5% for training and 1.38% for testing which are small compared to other reports in the literature. The small standard deviations and the small interval of less 3% between the training and testing accuracies show good generalization performance indicating that we have built a good model. We see an average $\mathbf{Q_3}$ testing accuracy of 81.67% while the individual testing accuracies are 79.08%, 75.58% and 90.37% respectively for the three classes. Similarly, the other metrics seen in this table such as specificity 90.12, sensitivity 68.67 and precision 81.67 are also higher compared to those found in the literature (not shown in the comparison table; when the final predictions are higher then, all other metrics also show correspondingly higher values). The average Matthew's correlation coefficient is 0.61.

We see the same trends here as we saw for the dataset-84 for the positive predictive value for β-sheet, which is 5% lower for α-helix and 6% lower for β-sheet, when prevalence is taken into consideration. There is no significant change for coil and the overall differences in accuracy fall by more than 4% when prevalence is taken into account for positive predictions. We see the opposite behavior for the negative predictive values with α-helix having the biggest gains and an overall 3% increase in accuracies, for negative values when prevalence is included.

On comparing the false positive (**FPR**) and false negative (**FNR**) rates, we see that there are almost 3 times more false negatives at 31.33% than false positives at 9.88%. The worst **FNR** seems to be for β-sheet followed by α-helix, while the accuracies seem to reverse themselves for **FPR**. The Likelihood positive values are over 8 indicating that the Post-test results yield a greater confidence for positive predictions than for negative predictions.

Results of a more recent study and information about the number of total and average residues and accuracies for the 5-fold cross validations are given in Table 4.6, 4.7, 4.8 and 4.9.

Table 4.5 Comparison of results for secondary structure predictions

This table compares the results of **FLOPRED** with some popular secondary structure prediction studies in the literature, which use the **CB513** dataset (except for PHD method). There are many other studies that give similar range of accuracies. Q_3 accuracies (C_H, C_E, C_C) of **FLOPRED** are seen to be 4.26% higher, compared to other studies given in this table. The correlation-coefficients are lower by 1% for α-helix but higher by 7% and 15% for β-sheet and coil respectively. These results are discussed further in Section 4.4.1 and 4.4.2

Method	**Data**	SOV%	**Q_3%**	**Q_H%**	**Q_E%**	**Q_C%**	C_H	C_E	C_C
PHD [1]		-	75.81	79.5	66.52	73.65	-	-	-
JNet [2]		74.21	76.4	78.4	63.9	80.6	-	-	-
PSIPRED [3]		76.0	79.69	81.41	71.59	78.15	0.75	0.69	0.63
SPINE [4]		-	80.0	84.4	72.2	80.5	-	-	-
MMBP [5]		-	85.6				-	-	-
Porter-H [6]		-	85.7				-	-	-
FLOPRED[7]		-	**89.96**	**88.74**	**87.75**	**93.39**	0.74	0.76	0.78
FLOPRED[8]		-	**85.16**	**88.20**	**85.04**	**82.23**	-	-	-

[1] (Rost and Sander, 1993)
[2] (Cuff and Barton, 2000)
[3] Jones (1999)
[4] (Dor and Zhou, 2007)
[5] (Yang et al., 2011)
[6] (Pollastri et al., 2007)
[7] dataset-84
[8] dataset-415

4.4.3 Comparative study with the literature

All the methods that are listed include multiple sequence alignments to develop their datasets whereas in our datasets we use sequence information and knowledge-based potential information calculated using the **CABS** algorithm. Our method provides accuracies which are higher than all the methods listed and provides for an improvement of 9.96% over the SPINE server (Dor and Zhou, 2007). Compared to the recent (Pollastri et al., 2007; Yang et al., 2011) methods which give an overall accuracy of 85.7% and 85.6%, our method for dataset-84 gives an improvement of 4.26% and it is almost the same for dataset-415 with a score of 85.16% which is lower by just 0.44%. The correlation-coefficients are lower by 1% for α-helix but higher by 7% and 15% for β-sheet and coil respectively, when compared to

PSIPRED.

4.5 Conclusions

Protein secondary structure predictions can be improved by including long range and short range interaction information gained from sequences and by using improved machine learning and optimization techniques. The increasingly available newer protein sequences could help with improved secondary structure predictions. At the same time the variety of homologous sequences can be a limiting factor since they provide varied information which makes it more complicated to build models with good generalization. Information gained from secondary structure predictions of proteins might lead to better understanding of the role of proteins in diseases and industrial applications and help to advance technology in these fields. A two stage Extreme Learning Machine approach was presented where an improved Neural Network algorithm called the Extreme Learning Machine was used for classifying a reduced set of three secondary structure, namely, alpha-helix, beta-strand and coil. The PSO algorithm was used to tune the **ELM** parameters in order to get higher classification results. The data was generated using **CATH** library (Orengo et al., 1997; Cuff et al., 2008) structures and **CABS** (Kolinski, 2004) force field. We were able to predict secondary structures with a higher training accuracy of 93.33% and a testing accuracy of 92.24% with a standard deviation of 0.48%, on a small group of 84 proteins, which shows good generalization performance. On a larger set of 415 proteins, we obtained a testing accuracy of 86.5% with a standard deviation of 1.38%. These results are much higher than those found in literature and are validated by a comparison of our results with similar studies.

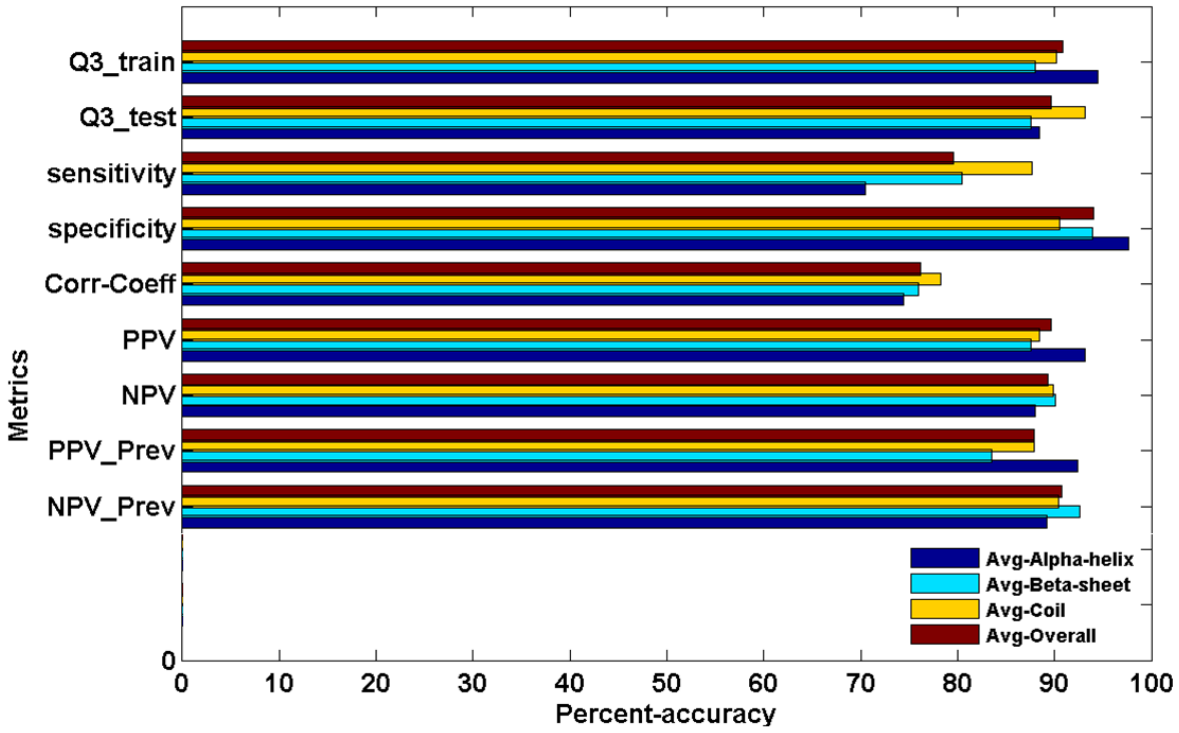


Figure 4.1 Metrics of the**testing** results for 84 proteins

This figure shows the $\mathbf{Q_3}$-training, $\mathbf{Q_3}$-testing, precision (positive predictive value or **PPV**), sensitivity, specificity and Matthew's correlation coefficient for a testing set of 2647 residues selected from a set of 84 proteins. These results are discussed further under Section 4.4.1 and Section 4.4.2. The data for this graph is given in Tables 4.2 and 4.4.

Table 4.6 Metrics of the **training** results for 415 proteins

This table shows the training results for each of the 5-fold cross validation models. The percentage values for the accuracies are shown on the right. These results are discussed further under Section 4.4.2.

Confusion Matrices - 5 fold CV						**Training accuracy %**		
		H	**E**	**C**	**Total**	**H**	**E**	**C**
CV-1	H	3030	21	400	3451	86.88	0.72	12.41
	E	4	1856	338	2198	0.21	81.20	18.59
	C	275	488	3582	4345	6.30	13.02	80.68
CV-2	H	3108	18	405	3531	84.91	1.10	13.99
	E	2	1880	329	2211	0.79	78.73	20.48
	C	271	498	3636	4405	7.06	13.69	79.24
CV-3	H	2888	16	372	3276	87.47	0.54	11.99
	E	3	1844	310	2157	0.23	80.83	18.94
	C	268	492	3507	4267	7.67	13.07	79.26
CV-4	H	2862	16	368	3246	86.19	0.82	12.99
	E	1	1860	327	2188	0.87	78.74	20.39
	C	253	471	3440	4164	8.09	13.88	78.03
CV-5	H	3311	14	401	3726	87.18	0.52	12.30
	E	1	1902	328	2231	0.27	82.26	17.47
	C	281	475	3304	4060	7.92	12.75	79.33

Table 4.7 Metrics for **testing** results for 415 proteins

This table shows the testing results for each of the 5-fold cross validation models. The percentage values for the accuracies are shown on the right. These results are discussed further under Section 4.4.2.

Confusion Matrices - 5 fold CV						**Testing accuracy %**		
		H	**E**	**C**	**Total**	**H**	**E**	**C**
CV-1	H	3151	26	450	3627	87.80	0.61	11.59
	E	6	2367	542	2915	0.18	84.44	15.38
	C	334	690	4275	5299	6.33	11.23	82.44
CV-2	H	3617	47	596	4260	88.02	0.51	11.47
	E	26	2591	674	3291	0.09	85.03	14.88
	C	408	791	4577	5776	6.15	11.31	82.54
CV-3	H	3903	24	535	4462	88.16	0.49	11.36
	E	5	1750	410	2165	0.14	85.49	14.37
	C	341	581	3523	4445	6.28	11.53	82.19
CV-4	H	4623	44	697	5364	88.17	0.49	11.34
	E	27	2440	632	3099	0.05	85.01	14.95
	C	434	744	4184	5362	6.08	11.31	82.61
CV-5	H	3529	21	498	4048	88.86	0.38	10.76
	E	7	2100	446	2553	0.04	85.25	14.70
	C	407	655	4075	5137	6.92	11.70	81.38

Table 4.8 Metrics for average **training results** for 415 proteins

This table shows the confusion matrices for the total and average number of residues used in the 5-fold cross-validation training models. The variation in the number of residues used for each model is given as a standard deviation. The training results and their standard deviation values are also given on the right for the 5-fold cross validation. These results are discussed further under Section 4.4.2.

Total number of residues - 5 CV						**Training accuracy %**		
		H	**E**	**C**	**Total**	**H**	**E**	**C**
	H	15199	85	1946	17230			
	E	11	9342	1632	10985			
	C	1348	2424	17469	21241			
Total					49456			
Avg-CV	H	3040	17	389	3446	86.52	0.74	12.74
	E	2	1868	326	2197	0.47	80.35	19.17
	C	270	485	3494	4248	7.41	13.28	79.31
Total					9891			
Avg-Std-dev	H	182	3	18	197	1.02	0.24	0.79
	E	1	23	10	28	0.33	1.57	1.27
	C	10	11	130	139	0.73	0.48	0.94
						Avg-acc		82.06
						Avg-Std-dev		1.18

Table 4.9 Metrics for average **testing results** for 415 proteins

This table shows the confusion matrices for the total occurrences and average number of residues used in the 5-fold cross-validation testing in each iteration. The variation in the number of residues used for each model is given as a standard deviation. The testing results and their standard deviation values are also given on the right for the 5-fold cross validation. These results are discussed further under Section 4.4.2.

Total number of residues - 5 CV						**Testing accuracy %**		
		H	**E**	**C**	**Total**	**H**	**E**	**C**
	H	3765	32	555	4352	88.20	0.50	11.30
	E	14	2250	541	2805	0.10	85.04	14.86
	C	385	692	4127	5204	6.35	11.42	82.23
Total					12361			
Avg-Std-dev	H	110	2	19		0.40	0.08	0.32
	E	2	66	23		0.06	0.39	0.37
	C	9	16	77		0.33	0.19	0.50
								85.16
								0.43

CHAPTER 5. AN AMINO ACID PERSPECTIVE OF SECONDARY STRUCTURE PREDICTION

[1]

5.1 Background and Significance

Most secondary structure predictions give the results of their prediction in terms of the three common secondary structures. It is rare to see an analysis of the results at the amino acid level with few exceptions. The results of classifications from several secondary structure prediction servers were studied and reanalyzed (Kazemian et al., 2007) to discern the patterns of prediction accuracies with respect to different amino acids. The authors gave an analysis of these results with an amino acid perspective based on the results of the servers. We have done an in-depth analysis of the amino acids accuracies obtained from our own results and see some interesting and intriguing patterns in the classification results. The set of proteins used are all globular and there are no membrane proteins included in the data set.

We have developed several new methods for secondary structure prediction and accuracy results are presented for the secondary structures α-helix, β-strand and coil for individual amino acid types. These results are obtained in our studies on secondary structure prediction using **FLOPRED** methodology on dataset-84 as discussed in Section 4.4.1 on page 86. We investigate the influence of the composition and physicochemical properties of amino acids in predicting secondary structures and see if there is a correlation between these properties and the prediction accuracies. These results might help to determine the influence of amino acids on formation of secondary structures themselves and may result in a deeper understanding

[1] In prepartion for submission to journal

of how these contribute to the final structure and functions of proteins.

5.1.1 Discussion of results for amino acid types in dataset-84

Table 5.1 $\mathbf{Q_3}$ test accuracies for amino acids in dataset-84

This table shows the $\mathbf{Q_3}$ accuracies for all amino acids in dataset-84. It also gives the ranking which is ordered from 1 (least accurate) to 20 (most accurate) for each of the amino acids. The results are discussed under Section 5.1.1.

	Rank					Amino acid
Amino Acid	**Alpha-helix**	**Beta-strand**	**coil**	**overall**	**Q_3-Acc**	**% Content**
Alanine	7	12	19	18	92.49	7.42
Arginine	9	3	7	3	86.70	5.37
Asparagine	4	6	12	7	88.00	3.86
Aspartic Acid	15	15	3	9	88.83	5.83
Cysteine	20	1	5	2	85.76	2.23
Glutamic acid	13	2	10	5	89.52	4.16
Glutamine	10	10	8	11	87.09	8.25
Glycine	5	7	13	8	88.40	7.57
Histidine	2	19	20	17	92.27	1.67
IsoLeucine	14	17	11	14	91.54	5.26
Leucine	8	9	16	13	90.40	8.21
Lysine	6	18	6	12	90.10	6.93
Methionine	19	14	18	20	93.84	1.40
Phenylalanine	12	13	2	6	87.54	3.63
Proline	1	4	4	1	82.29	4.88
Serine	18	5	9	10	89.19	5.60
Threonine	17	16	14	19	93.03	6.06
Trypotphan	3	20	15	15	91.86	1.10
Tyrosine	11	11	17	16	92.08	3.60
Valine	16	8	1	4	86.83	6.96

$\mathbf{Q_3}$ accuracies for individual amino acids for dataset-84 are shown in Table 5.1. We can see from the table that the highest accuracies are for Methionine, Threonine, Alanine, Histidine, Lysine, Leucine, Isoleucine, Tyrosine and Tryptophan whose accuracies are above 90% and whose rankings are above 14 (out of 20) while the lowest accuracies are seen for Proline, Cysteine, Arginine, Valine and Glutamic acid with rankings are below 6. The right-most column gives the percentage content of each amino acid in the test set. This table shows the amino

acid composition among the 2647 residues for each of the secondary structures in the test set. It can be seen that amino acids are not present in uniform quantities in each secondary structure. There are very few amino acids such as Glutamine, Glutamic acid and phenylalanine, which have comparable quantities in each of the three secondary structures. All the other amino acids are present in varying quantities in each of the structures. Proline, serine, threonine and glycine have the lowest number of α-helix residues while glutamic acid and leucine have the highest quantities. In the case of β-strand, aspartic acid, proline and serine seem to have the lowest content while isoleucine, phenylalanine, Threonine and valine have the highest content. Large numbers of residues are in coil structures for proline, asparagine, aspartic acid, glycine and serine, while valine, phenylalanine and glutamic acid seem to have the lowest presence in coil.

Table 5.1 shows $\mathbf{Q_3}$ accuracies for all amino acids in dataset-84. The accuracy ranking is given for each of the three secondary structures, where a rank of 1 is the lowest and a rank of 20 is the highest. The overall rank is given according to the final $\mathbf{Q_3}$ accuracies. The last column gives the percentage content of each of the amino acids. The values marked in red in the last two columns indicate the highest $\mathbf{Q_3}$ accuracies obtained for those particular amino acids. It can be observed that those with the highest accuracies are not always matched with the highest content in the last column. The content of each amino acid is given in sorted order in and Figure 5.7. In the cases of histidine (1.67%), methionine (1.4%) and tryptophan (1.1%) the content is the lowest compared to other amino acids while their accuracies are high, at between 92% and 94%. Some amino acids such as glutamic acid (8.25%) , glycine (7.57%) and valine (6.96%) have the highest content but have lowest accuracies between 87% and 88%. Figure 5.9 illustrates these values. It can be seen that proline has average content but the lowest accuracies at 82%. The overall accuracy is 89.38% with a standard deviation in the accuracies of 2.9%, which is low compared to the literature. It has always been argued that higher number of samples can lead to better accuracies but this is clearly not the case. It was proposed that one of the reasons for lower secondary structure accuracies over the years was due to lack of enough sequences which were representative of all the protein sequences. Here

we see that at least at the amino acid level, more content does not necessarily mean better accuracies for those amino acids.

5.1.2 Prediction accuracies and physicochemical properties

Table 5.2 shows five physicochemical properties of amino acids and the accuracies of amino acids that were obtained for dataset-84. It can be seen from this table that the residues that are hydrophobic have better accuracies than those which have other properties such as charge or polarity. Those categories such as size or rings (listed under special) do not necessarily enjoy higher accuracies, indicating that these properties may not be useful to discern secondary structure. Hence hydrophobic residues seems to be the one of the physicochemical properties among the five properties here, which enjoys higher accuracies, with the exception of phenylalanine and valine. Among the positively charged amino acids histidine and lysine enjoy higher accuracies, with the exception of Arginine. Residues with other properties seem to have average accuracies.

5.1.3 Prediction accuracies and content of amino acids in secondary structures

Figure 5.8 shows the ratio in which each amino acid is present in secondary structures. Some amino acids such as arginine and glycine are represented in equal proportions in all three secondary structures. Some amino acids such as valine, isoleucine and alanine are in unequal proportions but two of their accuracies are good at 92% . Tyrosine seems to have a large imbalance of content in the three secondary structures, yet it seems to have one of the highest accuracies of over 92%, while serine has an average accuracy of 89.2% even though its content is imbalanced among the three secondary structures. Histidine has a fairly even representation and a high accuracy of 92.3% while cysteine has a lower accuracy with imbalance in amino acid content in the three structures. So, there does not seem to be any significant correlation between uniform content and accuracies of secondary structure prediction.

5.2 Conclusions

The results of **FLOPRED** secondary structure have been analyzed with respect to their amino acid content. Our analysis indicates that high accuracies were obtained for some of the amino acids while reasonable contributions to accuracies came from other amino acids. There was not a great imbalance in the data with respect to any particular amino acids. A correlation study between content and accuracies revealed that there is no correlation between the content of an amino acid and its accuracy. In some cases, amino acids which comprised as little as 1% of the total content had the highest accuracies while some amino acids which were present in large quantities compared to others, had lower accuracies. This contradicts the perception that lack of data might be contributing to lower accuracies during classifications. Our studies with respect to the influences of the content of amino acids to secondary structure predictions can perhaps throw some light as on their influences on the formation of secondary structure and help to guide development of better future secondary structure prediction methods.

Table 5.2 Prediction accuracies and physicochemical properties

This figure shows five physicochemical properties of amino acids and the accuracies of amino acids that were obtained for dataset-84. It can be seen from this table that the residues which are hydrophobic have better accuracies than those which have other properties such as charge or polarity. This figure is further discussed under Section 5.1.2.

	Physical and chemical properties						
Amino Acid	**Special Property**	**Charged positive**	**Charged negative**	**Polar uncharged**	**Hydro-phobic**	**Q_3-Acc**	**% Amino acid**
Alanine					**Y**	92.49	7.42
Arginine		Y				86.70	5.37
Asparagine				Y		88.00	3.86
Aspartic Acid			Y			88.83	5.83
Cysteine	Y					85.76	2.23
Glutamic acid			Y			89.52	4.16
Glutamine				Y		87.09	8.25
Glycine	Y					88.40	7.57
Histidine		**Y**				92.27	1.67
IsoLeucine					**Y**	91.54	5.26
Leucine					**Y**	90.40	8.21
Lysine		**Y**				90.10	6.93
Methionine					**Y**	93.84	1.40
Phenylalanine					Y	87.54	3.63
Proline	Y					82.29	4.88
Serine				Y		89.19	5.60
Threonine				**Y**		93.03	6.06
Trypotphan					**Y**	91.86	1.10
Tyrosine					**Y**	92.08	3.60
Valine					Y	86.83	6.96

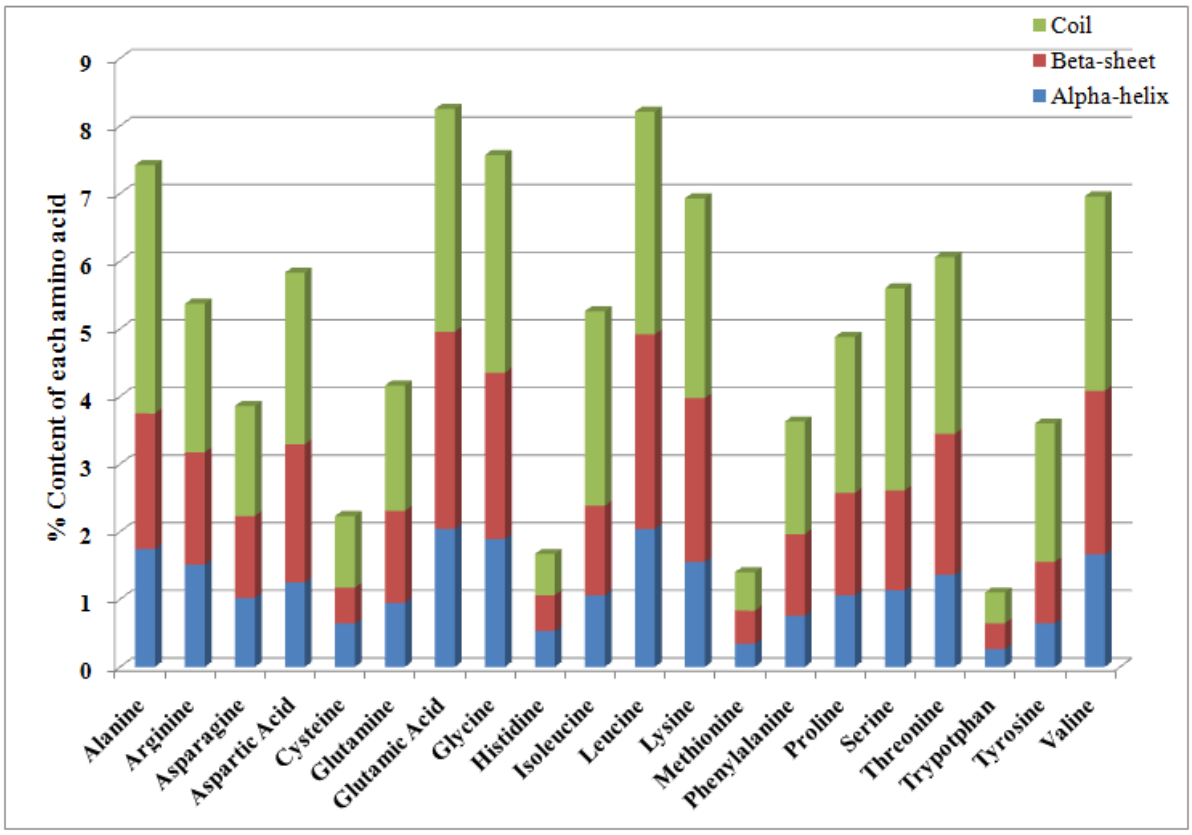


Figure 5.1 Ratio of amino acid content in secondary structures for dataset-84

This table shows the ratio in which each amino acid is present in secondary structures. Some amino acids such as arginine and glycine are represented in equal proportions in all three secondary structures. Some amino acids such as valine and alanine are in unequal proportions. The results are discussed under Section 5.1.1.

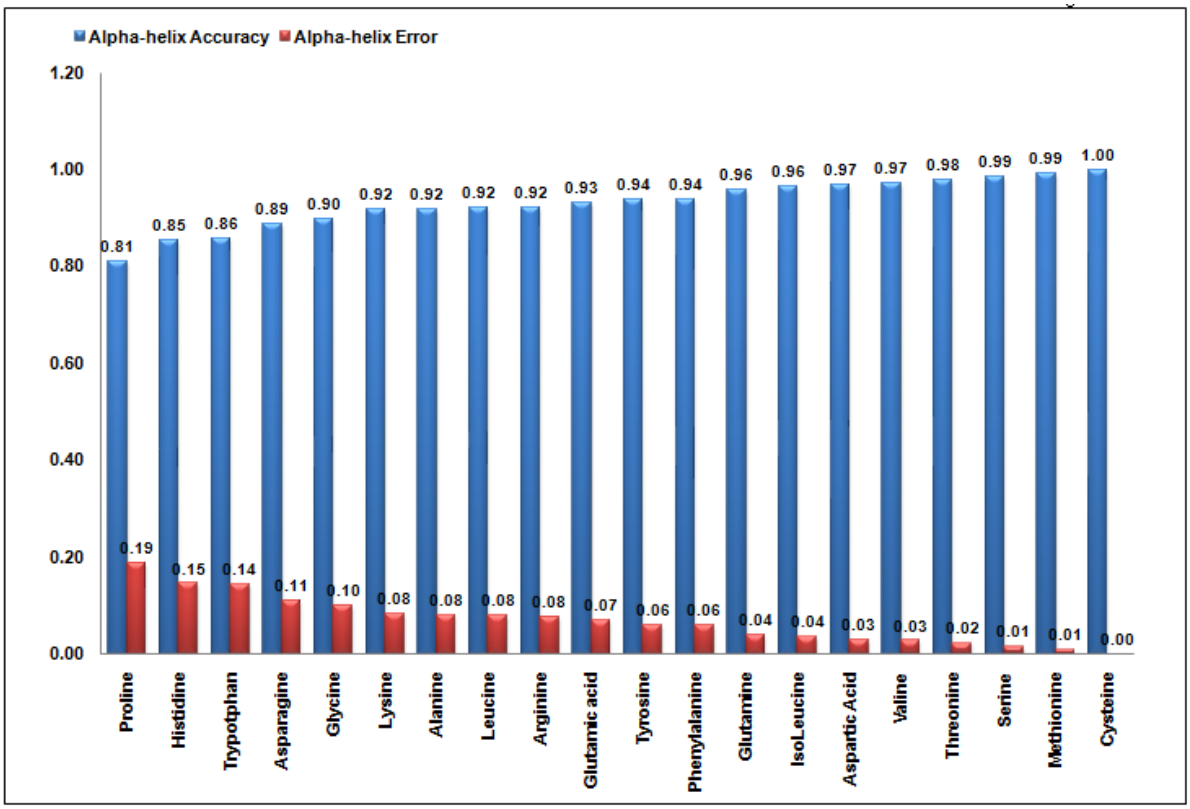


Figure 5.2 Test Accuracy and error in α-helix for all amino acids - dataset-84

This figure shows the accuracy and error in α-helices for each amino acid. The long bars give the accuracy while the shorter bars give the error rate. Data is included in this figure. These results are further discussed under Section 5.1.1.

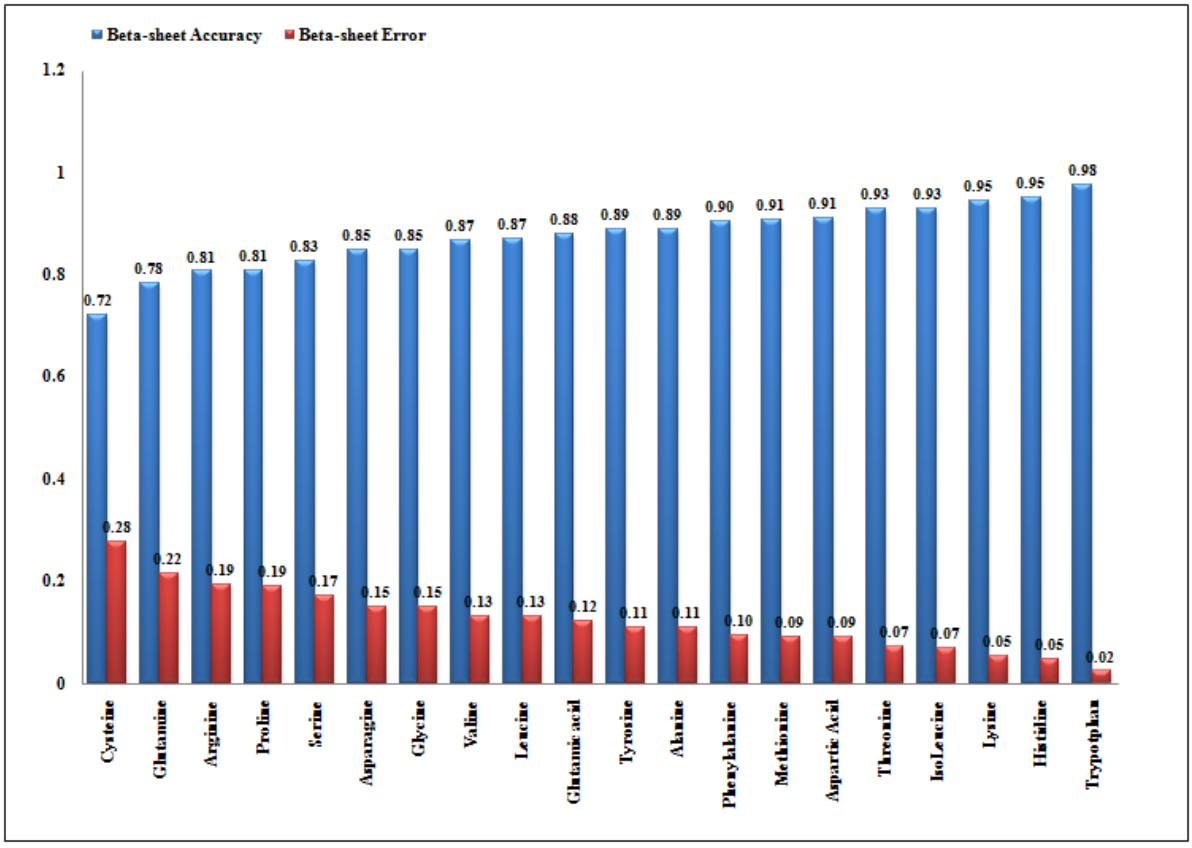


Figure 5.3 Test Accuracy and error in β-sheet for all amino acids - dataset-84

This figure shows the accuracy and error in β-sheets for each amino acid. The long bars give the accuracy while the shorter bars give the error rate. Data is included in this figure. These results are further discussed under Section 5.1.1.

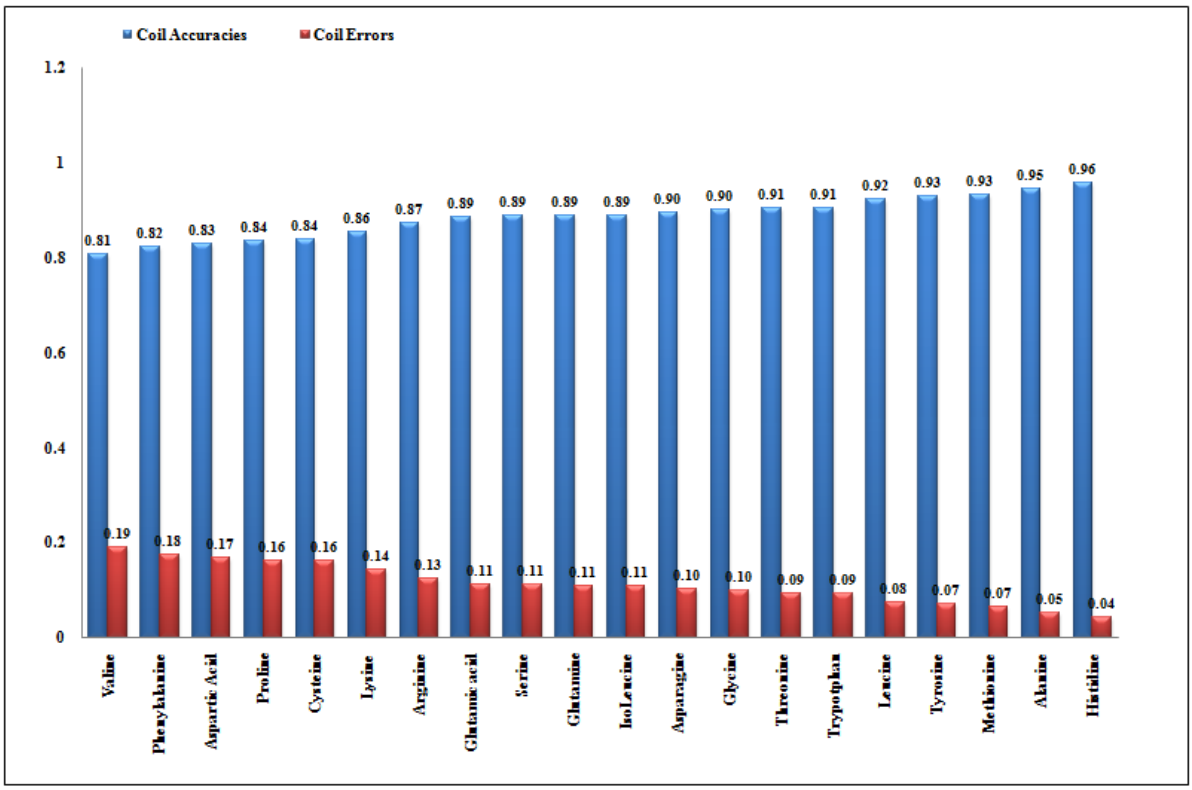


Figure 5.4 Test Accuracy and error in coil for all amino acids - dataset-84

This figure shows the accuracy and error in coils for each amino acid. The long bars give the accuracy while the shorter bars give the error rate. Data is included in this figure. These results are further discussed under Section 5.1.1.

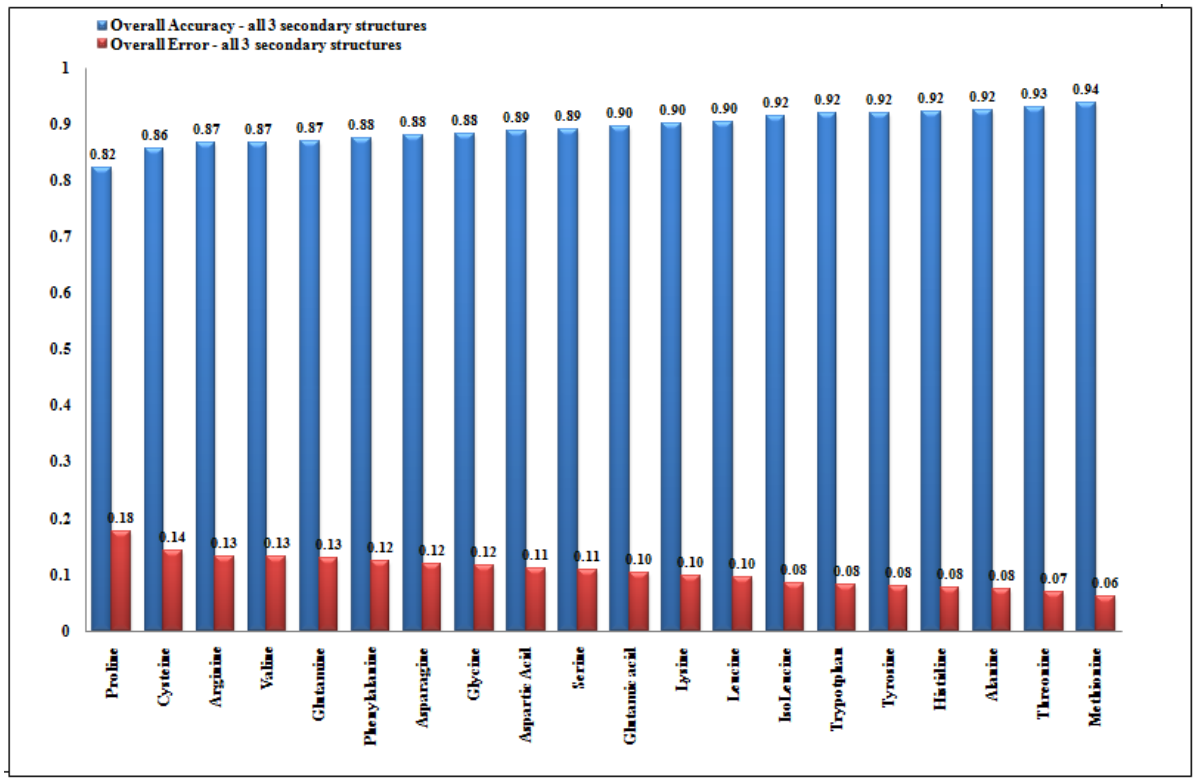


Figure 5.5 Overall test Accuracy and error for amino acids - dataset-84

This figure shows the overall accuracy and error for each amino acid. The long bars give the accuracy while the shorter bars give the error rate. Data is included in this figure. These results are further discussed under Section 5.1.1.

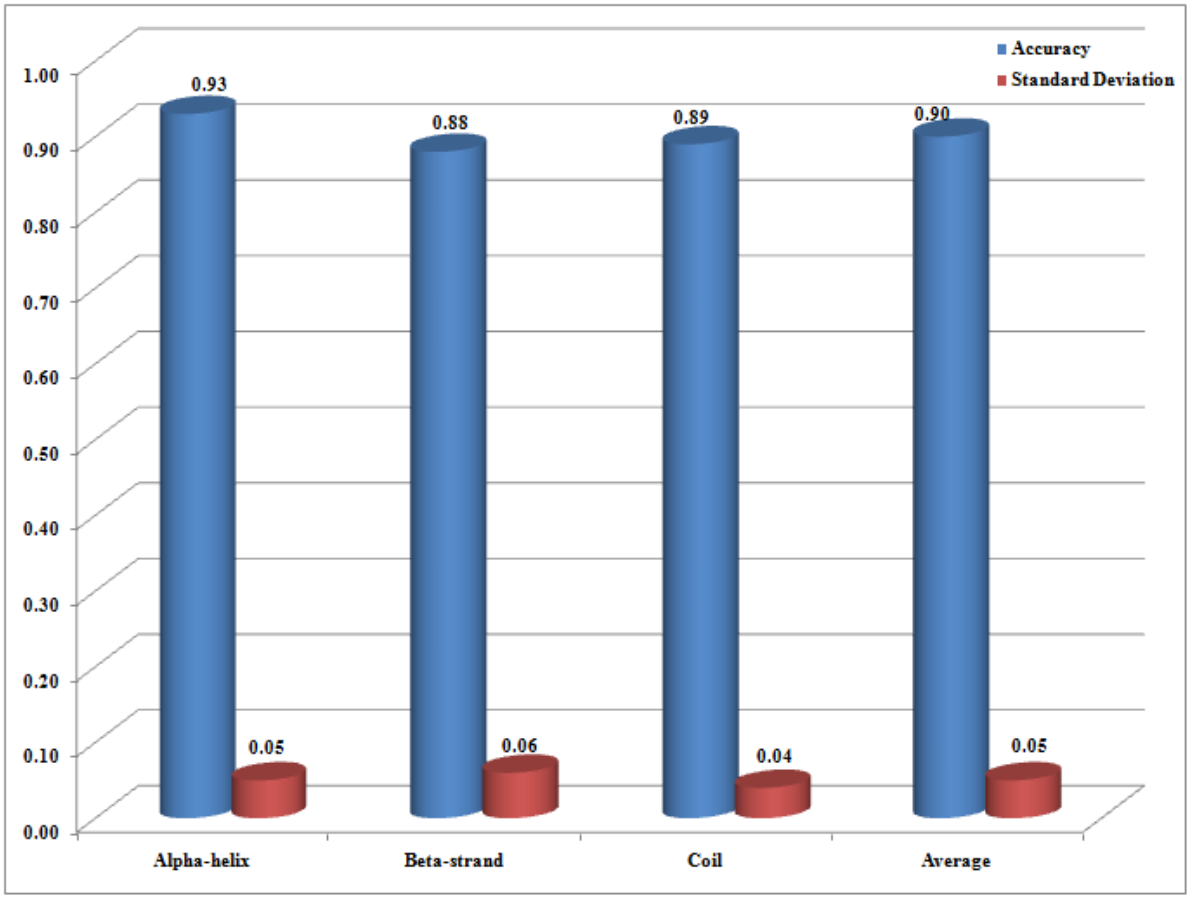


Figure 5.6 Overall test Accuracy and standard deviation for the three secondary structures - dataset-84

This figure shows the overall accuracy and standard deviation for the three secondary structures α-helix, β-sheet and coil and their average accuracy. The long bars give the accuracy while the shorter bars give the standard deviation. Data is included in this figure. These results are further discussed under Section 5.1.1.

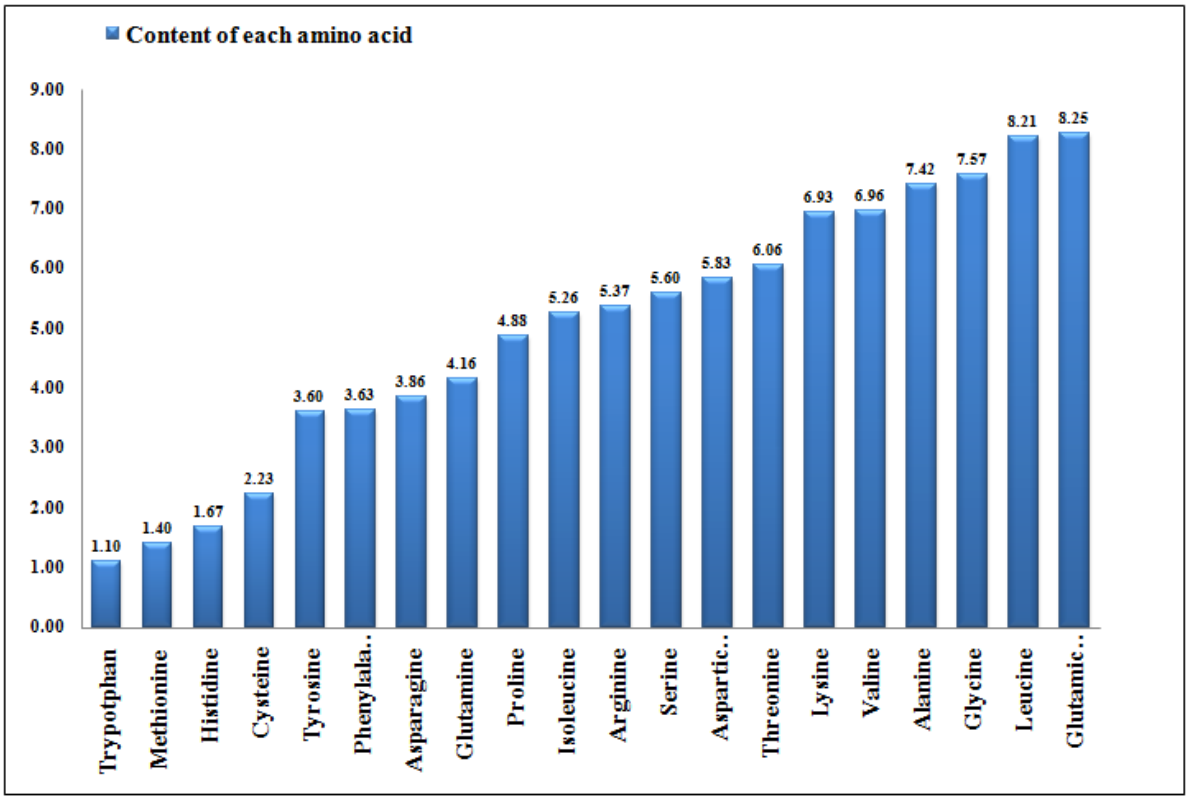


Figure 5.7 Sorted content of amino acids in the test set of dataset-84

This figure shows the amino acid composition among 2647 residues. It can be seen that certain amino acids like tryptophan, methionine, cysteine and histidine are present in very small quantities of 1% to 2% while others such as valine, alanine, glycine, leucine and glutamic acid are present in comparatively large quantities between 7% and 8% while some other amino acids are present in average quantities ranging between these two extreme values. The data for this figure is given under Table 5.1 . These results are further discussed under Section 5.1.1.

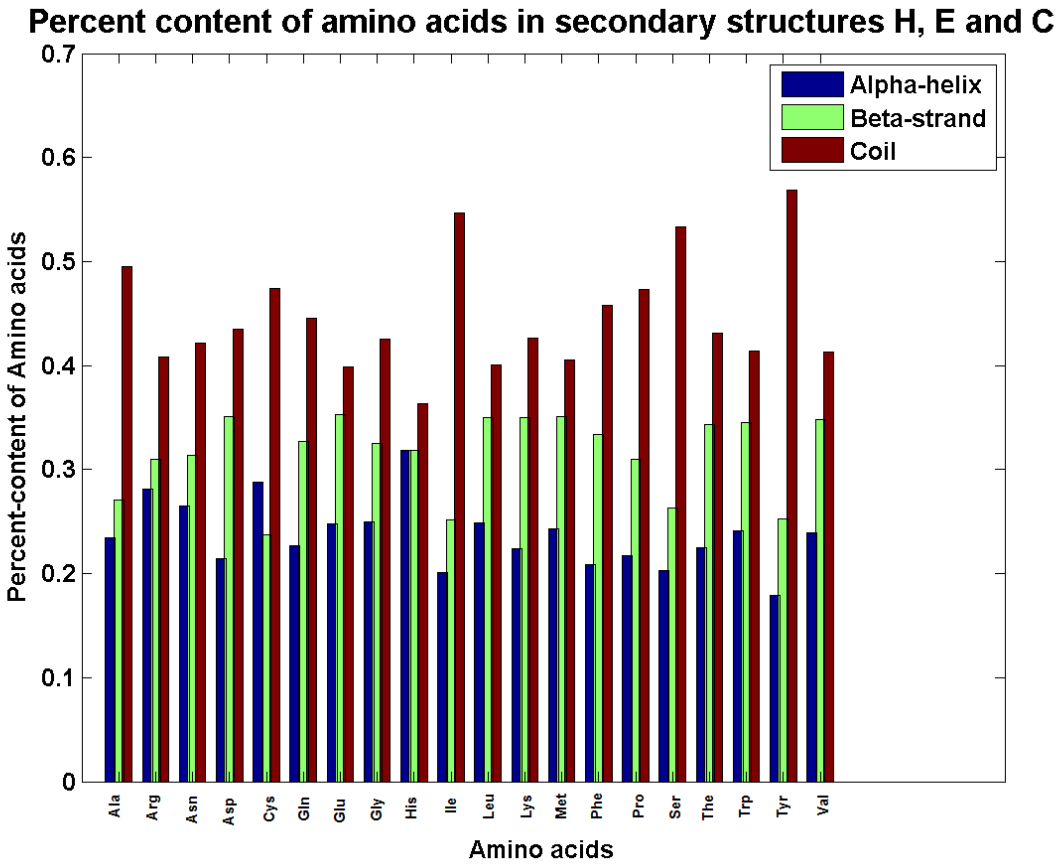


Figure 5.8 Content of amino acids in each of the three secondary structures in the test set of dataset-84

This figure shows the comparative content of amino acid composition among 2647 residues for each of the amino acids in the three secondary structures. The data for this figure is given under Table 5.7 . These results are further discussed under Section 5.1.1.

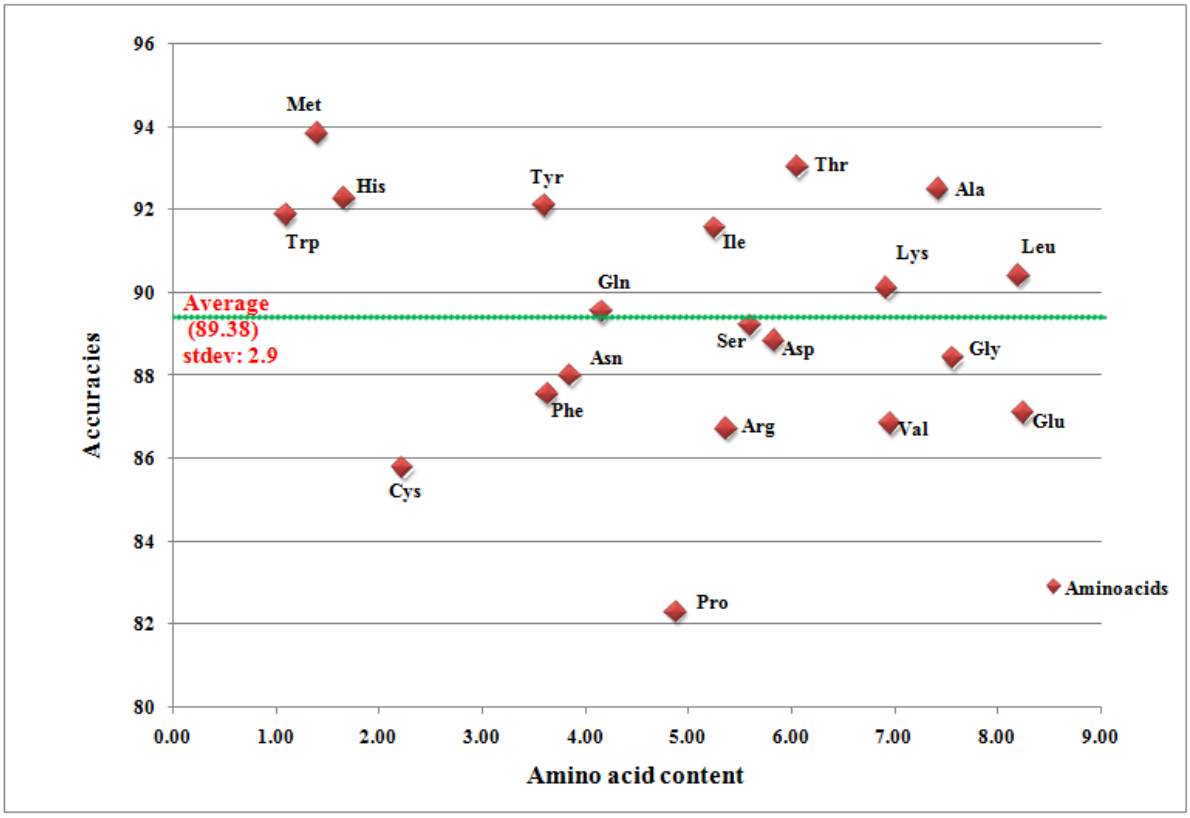


Figure 5.9 Correlation between accuracy and content for each amino acid in the test set of dataset-84

This figure shows the correlation between the content of the amino acids and their secondary structure prediction accuracies. It can be observed that there is no correlation since some amino acids such as methionine, histidine and tryptophan are present in low quantities but enjoy high accuracies which are above 90% while some amino acids such as glutamic acid, glycine and valine are present in comparatively large quantities and yet have lower accuracies which are between 86% and 88%. Proline has the lowest accuracy but average content in the data set. The data for this figure is given under Table 5.1 . This figure is further discussed under Section 5.1.1.

CHAPTER 6. FLOPRED FOR PROTEIN SECONDARY STRUCTURE PREDICTION USING PHYSICOCHEMICAL FEATURES OF AMINO ACIDS

Abstract

Protein secondary structure predictions, based on amino acid sequences, are commonly used as input to protein 3-D structure predictions. Physical properties of the constituent amino acids might influence the formation of particular secondary structures. Additional information such as the biophysical and chemical properties of amino acids might help to improve the results of secondary structure prediction. In order to determine the effects of the various properties of amino acids on the formation of protein secondary structures, a database of 544 physicochemical amino acid properties was used to encode protein sequences and this data was used for secondary structure prediction. Genetic Algorithm (GA) was used for feature selection and Principal Component Analysis (**PCA**) was used for feature reduction and **FLOPRED** was used for the predictions. **FLOPRED** methodology is a combination of a neural network based method called Extreme Learning Machine (**ELM**) and advanced Particle Swarm Optimization techniques. Preliminary studies using **FLOPRED** for secondary structure classification show promising results.

6.1 Introduction

Proteins consist of sequences of amino acid residues that play a key role in determining the secondary and tertiary structures of a protein. Various factors influence protein functions, such as the proteins' native structure, information encoded in its constituent amino acid sequences and its suitability in the surrounding environment when folded. All of these features

play an important role in protein function determination. Methods to predict protein structures occupy a central role in biological research due to their potential contributions to several important fields of study.

Proteins interact with their solvent environment and perform a variety of biological functions. These interactions depend on the chemical and physical nature of the amino acids in their sequences. It will be reasonable to assume that the physicochemical properties of the amino acids will determine, at least to some extent, the formation of secondary structures. Many studies in literature have used a few of these properties such as hydrophobicity, polarity, solvent accessibility and other common properties and included them as features in secondary structure predictions. Many secondary structure prediction methods have been used in the literature (Ooi et al., 1987; Shen and Vihinen, 2003; Adamczak et al., 2004; Cheng and Baldi, 2006; Saraswathi et al., 2010a). A two-step approach has been developed (Meshkin and Ghafuri, 2010) using feature selection on physicochemical properties of residues and Support Vector Regression (SVR) to predict RSA. Position specific residue preferences (PSRP) of amino acids that appear at the ends of secondary structures have been used (Richardson and Richardson, 1988; Duan et al., 2008) to improve secondary structure predictions accuracy by about 3%. A two-level mixed-modal support vector machine (**MMS**) was used (Yang et al., 2011) for secondary structure prediction using physicochemical properties of amino acids and position-specific scoring matrices (**PSSM**) generated from **PSI-BLAST** (Altschul et al., 1997b) to achieve accuracies of up to 85.6%, the highest accuracy seen so far. We have developed new sets of features using physicochemical properties of amino acids from the **AAindex** database and use this data for secondary structure prediction. The data and methods used are discussed next.

6.2 Data and Methods

6.2.1 Data generation - Encoding physicochemical properties

A database was set up where protein sequences from the **CB513** data set (Cuff and Barton, 2000) were encoded using the physicochemical properties of amino acids derived from

the AAindex (Kawashima et al., 1999) database. AAindex is a database of amino acid physicochemical properties that were collected from previous publications. Some of the properties that are indexed in this database include amino acid chemical shifts, **RSA** values, molecular weights, lengths of side chains, hydrophobicity, volume, flexibility, amphiphilicity, frequency and so many other properties. In total 544 properties of amino acids were normalized to values between 0 and 1 and stored. The color map of these values is shown in Figure 6.1 where red is a high value and blue is a low value closer to zero. The varying colors represent the varying values for different amino acid physicochemical properties, which may help machine learning algorithms find good patterns in the sequence data when these values are encoded for the sequences. In principle, good patterns can lead to higher accuracies for secondary structure predictions. If the sequences were to be represented with orthogonal binary values, these rich patterns would not be available since 95% of the values will be zeros and only 5% of the values will be a 1, resulting in an almost complete blue colored values (zero) with occasional red dots (for ones). The values from this colorful matrix (Figure 6.1) were then encoded into sequences using a moving window of 9 residues. Each amino acid in the window is then substituted with its corresponding values from the stored table of physicochemical properties. This process was repeated 544 times, once for each amino acid property. This resulted in 4896 ($544 * 9$) features for each residue of interest in the fifth (middle) position in a window of 9 residues. A sample encoding of protein 1ahb2 is given in Figure 6.2. It would be very computationally intensive to use these large feature sets for secondary structure predictions. It will also need a lot of computational resources in terms of memory and computing power. Hence the features were reduced by using two different methods. One of the methods used is a *feature selection method*, where a Genetic Algorithm is used. The desired number of features can be specified and the algorithm will go through all the features and select the best set which will maximize prediction accuracy. We can select as many as 500 properties or as few as 20 properties and determine which set of values yield the best classification accuracies. The classification accuracies are evaluated using our **FLOPRED** methodology. The second method is a *feature reduction method* where Principal Component Analysis (**PCA**) is used to

reduce the data set into its most important components. Each of these reduced data sets were then used for secondary structure prediction. A preliminary study has been conducted using these two methods. There are large amounts of data that need to be processed and evaluated before definitive results can be discussed with respect to the usefulness of all 544 properties for predicting secondary structures. Our preliminary results are given under the results section. A description of the Genetic Algorithm and Principal Component Analysis used for this study is given next.

6.2.2 Integer coded Genetic Algorithm (ICGA) for Gene Selection

The genetic algorithm (**GA**) is perhaps the most well-known of all evolution-based search techniques. Genetic algorithms, which are based on evolutionary search techniques (Goldberg, 1989; Holland, 1975; Michalewicz, 1996), were developed in an attempt to explain the adaptive processes of natural systems and to design artificial systems based upon these natural systems. Genetic algorithms are widely used to solve complex optimization problems where the number of parameters and constraints are large and analytical solutions are difficult to obtain. In recent years, many schemes for combining genetic algorithms and neural networks have been proposed and tested for feature selection (Suresh et al., 2010; Saraswathi et al., 2011). In this study we have used the **GA** algorithm to select the best features (which encode protein physicochemical properties) which are likely to make the maximum contributions for secondary structure prediction.

Genetic Algorithms model evolution at the gene level. The components of the **GA** consist of *String Representation, Selection Function, Genetic Operators and the Fitness Function*. **GAs** use representations of fixed length strings. String representation is the process of encoding a potential search node (solution) as a string. In **GA**, string representation depends on the structure of the problem and on the genetic operators used in the algorithms. In earlier work on genetic algorithms (Holland, 1975; Goldberg, 1989), the string values were restricted to binary digits (0 and 1). A natural number (positive) representation of strings is considered to be more efficient (Michalewicz, 1996) and hence produces better results. Hence, in our studies,

we have used an Integer Coded Genetic Algorithm **(ICGA)** (Saraswathi et al., 2011) in which the string representation for search nodes is encoded as a string of M independent integers, where M represents the length of the string. There are three main processes essential for accurate functioning of Genetic Algorithms; *selection, cross-over and mutation*. **GA** is a search algorithm based on the mechanism of natural selection that transforms a set of individuals (population of fixed length strings) into a new population (i.e., the next generation) using genetic operators such as crossover and mutation (Holland, 1975; Goldberg, 1989; Michalewicz, 1996). These processes are generated similarly to what occurs in our genes. For each action, a fitness value is assigned and the features that give the *fittest* values will finally survive after going through several iterations (Michalewicz, 1996). A survival of the fittest strategy is adopted to identify the best strings and subsequently the genetic operators are used to create the next generation. Genetic algorithms have been successfully used to obtain solutions for many combinatorial optimization problems.

Genetic algorithms applied to combinatorial optimization problems work as follows: The search space contains all the search nodes for the given combinatorial optimization problem. **GA** starts with an initial population of N search nodes from the search space. Each search node in the population is evaluated using the objective function and the fitness is assigned to each search node. New search nodes are generated for the next generation based on the fitness value and by applying genetic operators (crossover, mutation and reproduction) to the current search nodes. This process is continued for several generations until the algorithm converges. The Genetic Algorithm uses the concept of survival of the fittest by passing *good* search nodes to the next generation, and combining different search nodes to form new generations.

We need to address the following factors in order to apply **ICGA** for selecting the best features from a given data set. Only features selected through this process will be used in the classification models used for further studies. Each of these factors is discussed next.

6.2.2.1 String Representation

In our studies, the string representation for a search node is encoded as a string of independent integers, where M represents the length of the string. The integer values in the string represent the selected features from the given set of features. For example, the string 1,10,22, 25,35,42,47,48,49,50 represents a set of 10 independent features selected from an initial set of 50 features. In this string representation of search nodes, we can uniquely represent all combinations of the M best features in the given set and use only these features for our classification model.

6.2.2.2 Population Initialization

In **GA**, an initial population of N search nodes is generated using a random selection from the given set of features. The size of population N and the method of initialization will affect the convergence of the problem. Since **GA** can iteratively improve the classification accuracy, the initial population can start off with an existing solution or it can potentially be a good solution by itself. Future populations can be randomly generated and used to improve the existing solution. The population size N is typically problem-dependent and has to be determined through simulations.

6.2.2.3 Selection Function

In **GA**, the selection of a search node plays an important role. This node is selected from existing search nodes (population), in order to produce new search nodes for the ensuing generations. A probabilistic selection is performed using genetic operators, where this selection is based upon the fitness of search nodes, such that the better search nodes have a better chance of being selected for producing new search nodes. It is possible that a search node in the population can be selected more than once for producing new search nodes, but we ensure that the final set of nodes will be unique. In the literature (Goldberg, 1989; Michalewicz, 1996), there are several schemes such as roulette wheel selection and its extensions, scaling techniques, tournament, elitist models and ranking methods which are presented for the se-

lection process. In our study, we have used the normalized geometric ranking method given by Michalewicz (1996) for the selection process. Here, the search nodes in the population are arranged in decreasing order of their fitness values, and a rank is assigned to each of the search nodes. The ranking method assigns a probability of selection, s_j to each search node j, based on its rank in the partially ordered set. Let q be selection probability for selecting the best search node and r_j be the rank of the j^{th} search node in the partially ordered set. The probability s_j of search node j being selected, using normalized geometric ranking method is

$$s_j = q^{'}(1-q)^{r_j-1} \tag{6.1}$$

where $q^{'} = \frac{q}{1-(1-q)^N}$ and N is the population size.

6.2.2.4 Genetic Operators

Genetic operators provide the basic search mechanism of the **GA**. The operators are used to create new search nodes based on existing search nodes in the population. Two types of operators namely crossover and mutation are commonly used in **GA**. Crossover is the primary operator in **GA**, and mutation is a secondary operator. Reproduction is another genetic operator. The genetic operators for feature selection problems are described below.

- **Crossover operation**: Crossover operation uses two search nodes (parents) to produce two new search nodes (off-springs). During the crossover, the parents exchange parts of their solutions (search nodes). This is done in order to combine and pass on parts of the good solutions present in each parent to produce the off-springs. In this work, a heuristic crossover operator is used to generate valid solutions (Michalewicz, 1996), where the fitness values of the two parent chromosomes are used to determine the direction of the search.

- **Mutation operation**: The mutation operation alters one search node (solution) to produce a new search node. Mutations introduce a certain amount of diversity into the population and are also useful to overcome premature convergence and local minima

problems. In this operation, the mutation site is selected randomly and a new value is assigned to the site based on a random integer generated from the range of numbers representing the feature set. This new number must be different from the feature numbers already present in the string.

Let U be the search node selected for mutation and the mutation site is shown in bold face.

$$U = [1, 5, 11, 14, \mathbf{24}, 30, 33, 40, 47, 50] \tag{6.2}$$

The feature 24 is removed and a new feature (3) different from those features already present in the string U is inserted. The string generated after the insertion operation will then be:

$$U = [1, 5, 11, 14, \mathbf{3}, 30, 33, 40, 47, 50] \tag{6.3}$$

- **Reproduction operation**: Reproduction is a commonly used genetic operator. We have used an elitist model discussed in Michalewicz (1996) for our experiments. In this method, the best search node (solution) generated in the current generation is passed on to the population in the next generation.

6.2.2.5 Fitness Function

The fitness function in genetic algorithms is typically the objective function that we want to optimize for the given problem. Our objective is to select the best M features that are required to develop a classification model. The performance of the classification model is evaluated using a different set of samples other than those used for model development. In this study, we use a single hidden layer neural network called *Extreme Learning Machine* for classifier model development. The **ELM** network is developed using a set of training samples and its performance is tested on a set of independent samples. The classification accuracy (η) obtained from the testing samples is used as the fitness value.

$$F = \eta \tag{6.4}$$

6.2.2.6 Termination Function

In GA, for each generation, solutions are selected on the basis of their fitness and are subject to genetic operations such as crossover and mutation. The evolution process of successive generations continues until a termination criterion is satisfied. The most frequently used stopping criteria are population convergence and specification of a value indicating maximum number of generations. The population convergence criterion used for our problem is specified as the incident when the solutions in the population are the same for two successive generations. We have used this as the termination criterion. If this does not occur then the maximum number of generations will be the stopping criteria.

6.2.3 Efficacy of the Integer Coded Genetic Algorithm

In previous studies using this algorithm, we have seen that the genetic algorithm has the capability of selecting optimal features that contribute to higher accuracies in classification (Saraswathi et al., 2011). In this study, this algorithm is used for secondary structure prediction, using data that encode the physiochemical properties of amino acids. Preliminary results are promising and we hope to improve it further.

6.2.4 Principal Component Analysis

Global optimization methods such as Particle Swarm Optimization can be applied to a given data set when the number of features are within manageable limits. When the feature set is very large such as microarray data where there are thousands of features, it is necessary to reduce the number of features using some feature reduction methods such as Principal Component Analysis (**PCA**). In our studies relating to structural effects of over 544 physicochemical properties of amino acids in protein sequences, a database has been developed using a moving window of 9 residues centered on each amino acid of interest. This results in a set of 4896 ($9 * 544$) features for each amino acid representation, that needs to be reduced using PCA, before the data can be processed by **FLOPRED** for secondary structure prediction. A simple methodology is proposed for secondary structure prediction, to perform sampling in

high dimensional spaces through the combined use of a family of particle swarm optimizers and reduction techniques, using PCA.

Generally, multiple evaluations of the objective (or fitness) function are carried out to obtain an optimal result. A large number of features will result in costly forward evaluations and hamper the use and effectiveness of global optimization algorithms such as **PSO**. Sampling can be performed in a reduced model space to obtain results close to the optimum. Dimension reduction is accomplished by **PCA** computed on a the reduced data set using stochastic simulation techniques. The use of a reduced basis helps to regularize the problem and to find a set of equivalent models that fit the data within a prescribed tolerance, allowing analysis of the data around the minimum misfit solution obtained using FLOPRED. **PSO** was chosen for optimization because its shows interesting exploration and exploitation capabilities, as discussed in previous chapters.

The reduction of the existing features to a set of basis vectors (that are consistent with our prior knowledge of amino acid properties) allows us to reduce the space of possible solutions. The data reduced to a set of bases using **PCA** should have the following desirable properties:

- The bases can be ranked, and allow classification of the model variability.
- The bases are orthonormal and allow us to take into account the contribution of each model parameter independently in order to reduce the bases.
- The bases are separable (enable good classification of the data) and enable us to expand our methodology to higher dimensions.

Principal component analysis (Pearson, 1901) is a well-known mathematical procedure that transforms a number of correlated variables into a smaller number of uncorrelated variables called principal components. The resulting transformation is such that the first principal component accounts for much of the variability and each succeeding component accounts for less of the remaining variability (Jolliffe, 2002). **PCA** involves finding orthogonal bases of the experimental covariance matrix estimated using the available data with a large number of features. We then select a subset of the most important principal components that are used

as the reduced model bases. The cloud versions of the different **PSO** optimizers discussed earlier in Section 1.4.2 on page 11 are used on these basis vectors, for secondary structure prediction. Preliminary results are promising.

If we have a matrix of models $X(n, l)$ where n is the dimension of the model space (or data space) and l is the number of samples that you have in the model space. Then the $\mathbf{C}_{prior}$ = $(\mathbf{X}-\mu)*(\mathbf{X}-\mu')$ is the transposed centered matrix. $\mathbf{C}_{prior}$, which is of size (n, n), is symmetric and semi definite positive so it admits orthogonal diagonalization as follows:

$\mathbf{C}_{prior}$ = VDV' where columns of V are the eigenvectors and D contains the eigenvalues .

The **PCA** method used is described as follows:

1. Initially, we generate an ensemble X = [$\mathbf{m}_1$, $\mathbf{m}_2$, . . . , $\mathbf{m}_q$] of plausible scenarios that are constrained using the prior information encoded in the data.

2. We need to find a set of patterns {$\mathbf{v}_1$, $\mathbf{v}_2$,, . . . , $\mathbf{v}_q$} that provide an accurate lower dimensional representation of the original set with q being much smaller than the dimension of the model space.

   PCA does this by diagonalizing the prior experimental covariance matrix:

   $\mathbf{C}_{prior}$ = $\frac{1}{N}\sum_{k=1}^{N}(\mathbf{m}_k - \mu)\,(\mathbf{m}_k - \mu)^t$

   where μ = $\frac{1}{N}\sum_{k=1}^{N}\mathbf{m}_k$ is the experimental ensemble mean.

   This ensemble covariance matrix is symmetric and semi-definite positive, hence, diagonalizable with orthogonal eigenvectors $\mathbf{v}_k$, and real semi-definite positive eigenvalues. Eigenvectors $\mathbf{v}_k$ are called principal components, the d first eigenvectors representing most of the variability in the model ensemble. Then, any model in the reduced space is represented as a unique linear combination of the d first eigenmodels $\mathbf{m} = \mu + \sum_{k=1}^{d} a_k\mathbf{v}_k$.

6.3 Results and discussion

A preliminary study has been carried out on a small number of 30 proteins, where 2000 residues were encoded with the **AAindex** feature values. A **PCA** toolbox (Fernández-Martínez

et al., 2010) was used to reduce 4896 features down to 120 features and a secondary structure classification was done on this data using **FLOPRED**. (Since this is a preliminary study, this algorithm has not yet been optimized by fully using advanced PSO). On this data with reduced features, secondary structure prediction accuracy was 82% for training and 65% on testing. These tests were averaged over 25 runs and have a standard deviation of 7% and 9% for training and testing respectively, as seen in Table 6.1.

The same set of data with 2000 residues and 4896 features were used for feature selection by the Genetic Algorithm. Only about 180 (20 * 9) features were selected, in sets of 9 features (window-size) so as to make sure that all columns belonging to a particular property were all included in the feature selection). These 180 features were used for secondary structure classification, and we obtained a training accuracy of 79% and testing accuracy of 70%. These results were averaged over 15 runs and they have a standard deviation of 10% and 3% for training and testing, respectively, as seen in Table 6.2. It can be seen that the training and testing accuracies vary widely with a standard deviation of 10% and 3% respectively. Although it will be useful to look at the selected features to learn what are the important features, we have not investigated this because of the size of the data. Ultimately we hope to find the best set of features that will give good secondary structure predictions when we repeat this study on on a larger data set.

The approach needs to be tested on a larger, newer protein data sets. In addition to this, the results need to be optimized using the advanced **PSO** algorithms that were used in other studies to give optimal results. The current studies are preliminary runs to see if the AAindex data could give good secondary structure prediction accuracies. These data are encoded only with AAindex data and no multiple sequence alignments or PSSM values have been used. Yet the results we have seen are promising and we plan to apply these algorithms on larger data sets in our future studies.

We also tried a combination of **PCA** and **GA**, where the **GA** was used to select the features and **PCA** was used to reduce this set further. The **FLOPRED** methodology was used on this data for secondary structure prediction. This method did not work very well (yielded only

53% accuracy), probably due to the fact that the features were selected based on a very small set of proteins.

6.4 Conclusions

A small set of proteins was used to encode 544 physicochemical properties of amino acids. The sequences were coded using a window of 9 residues to obtain a total of 4896 features. These features were reduced using **GA** and **PCA** to obtain secondary structure prediction accuracies that look promising. Future studies will include larger protein sets and advanced **PSO** techniques. We hope to find the best set of amino acid properties which contribute most to secondary structure prediction.

Table 6.1 Accuracy for **PCA** reduced features of AAindex properties.

This figure shows the accuracies for secondary structure prediction on a small set of 30 proteins encoded with 4896 features from AAindex data which were reduced to 120 features using Principal Component Analysis.

Testcase	**Training**	**Testing**
Test 1	0.85	0.76
Test 2	0.69	0.74
Test 3	0.84	0.69
Test 4	0.85	0.76
Test 5	0.74	0.63
Test 6	0.89	0.82
Test 7	0.92	0.75
Test 8	0.84	0.76
Test 9	0.86	0.57
Test 10	0.88	0.68
Test 11	0.86	0.64
Test 12	0.74	0.58
Test 13	0.79	0.60
Test 14	0.78	0.44
Test 15	0.89	0.60
Test 16	0.80	0.71
Test 17	0.70	0.64
Test 18	0.91	0.64
Test 19	0.84	0.58
Test 20	0.71	0.58
Test 21	0.81	0.56
Test 22	0.90	0.63
Test 23	0.83	0.61
Test 24	0.74	0.63
Test 25	0.81	0.55
Average	0.82	0.65
Std-dev	0.07	0.09

Table 6.2 Accuracy for **GA** selected features of AAindex properties.

This figure shows the accuracies for secondary structure prediction on a small set of 30 proteins encoded with 4896 features from **AAindex** data. Genetic Algorithm was used to select the best set of 120 features from this larger set. These reduced set of 120 features were then used for secondary structure prediction.

Testcase	**Training**	**Testing**
Test 1	0.81	0.70
Test 2	0.63	0.69
Test 3	0.69	0.73
Test 4	0.95	0.66
Test 5	0.70	0.75
Test 6	0.89	0.64
Test 7	0.74	0.69
Test 8	0.86	0.70
Test 9	0.81	0.67
Test 10	0.81	0.67
Test 11	0.91	0.73
Test 12	0.75	0.73
Test 13	0.83	0.69
Test 14	0.68	0.75
Test 15	0.56	0.70
Average	0.79	0.70
Std-dev	0.10	0.03

REFERENCES

Adamczak, R., Porollo, A., and Meller, J. (2004). Accurate prediction of solvent accessibility using neural networks-based regression. *Proteins*, 56:753–67.

Cheng, J. and Baldi, P. (2006). A machine learning information retrieval approach to protein fold recognition. *Bioinformatics*, 22:1456–1463.

Cuff, J. A. and Barton, G. J. (2000). Application of multiple sequence alignment profiles to improve protein secondary structure prediction. *Proteins*, 40:502–511.

Duan, M., Huang, M., Ma, C., Li, L., and Zhou, Y. (2008). Position-specific residue preference features around the ends of helices and strands and a novel strategy for the prediction of secondary structures. *Protein Science*, 17:1505–1512.

Fernández-Martínez, J. L., Mukerji, T., and García-Gonzalo, E. (2010). Particle swarm optimization in high dimensional spaces. *Swarm Intelligence*, 77:496–503.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, New York.

Holland, H. J. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer.

Kawashima, S., Ogata, H., and Kanehisa, M. (1999). AAindex: Amino Acid Index Database. *Nucleic Acids Research*, 27:368–369.

Meshkin, A. and Ghafuri, H. (2010). Prediction of Relative Solvent Accessibility by Support Vector Regression and best-first method. *Experimental and Clinical Sciences*, 9:29–38.

Michalewicz, Z. (1996). *Genetic Algorithm + Data Structures = Evolution Programs*. Springer-Verlag, Berlin, Heidelberg.

Ooi, T., Oobatake, M., Namethy, G., and Scheraga, H. (1987). Accessible surface areas as a measure of the thermodynamic parameters of hydration of peptides. *Proceedings of the National Academy of Sciences of the United States of America*, 84:3086–3090.

Pearson, K. J. (1901). Principal Components Analysis. *The London, Edinburgh and Dublin Philosophical Magazine and Journal*, 6:566.

Richardson, J. S. and Richardson, D. C. (1988). Amino acid preferences for specific locations at the ends of alphahelices. *Science*, 240:1648–1652.

Saraswathi, S., Jernigan, R. L., and Kloczkowski, A. (2010). An Extreme Learning Machine Classifier for prediction of relative solvent accessibility in proteins. *Proceedings of IJCCI/ICNC*, pages 364–369.

Saraswathi, S., Suresh, S., and Sundararajan, N. (2011). Icga-pso-elm approach for accurate multiclass cancer classification resulting in reduced gene sets in which genes encoding secreted proteins are highly represented. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 8:452–463.

Shen, B. and Vihinen, M. (2003). RankViaContact: ranking and visualization of amino acid contacts. *Bioinformatics*, 19:2161–2162.

Suresh, S., Saraswathi, S., and Sundararajan, N. (2010). Performance enhancement of extreme learning machine for multi-category sparse cancer classification. *Engineering Applications of Artificial Intelligence*, 23:1149–1157.

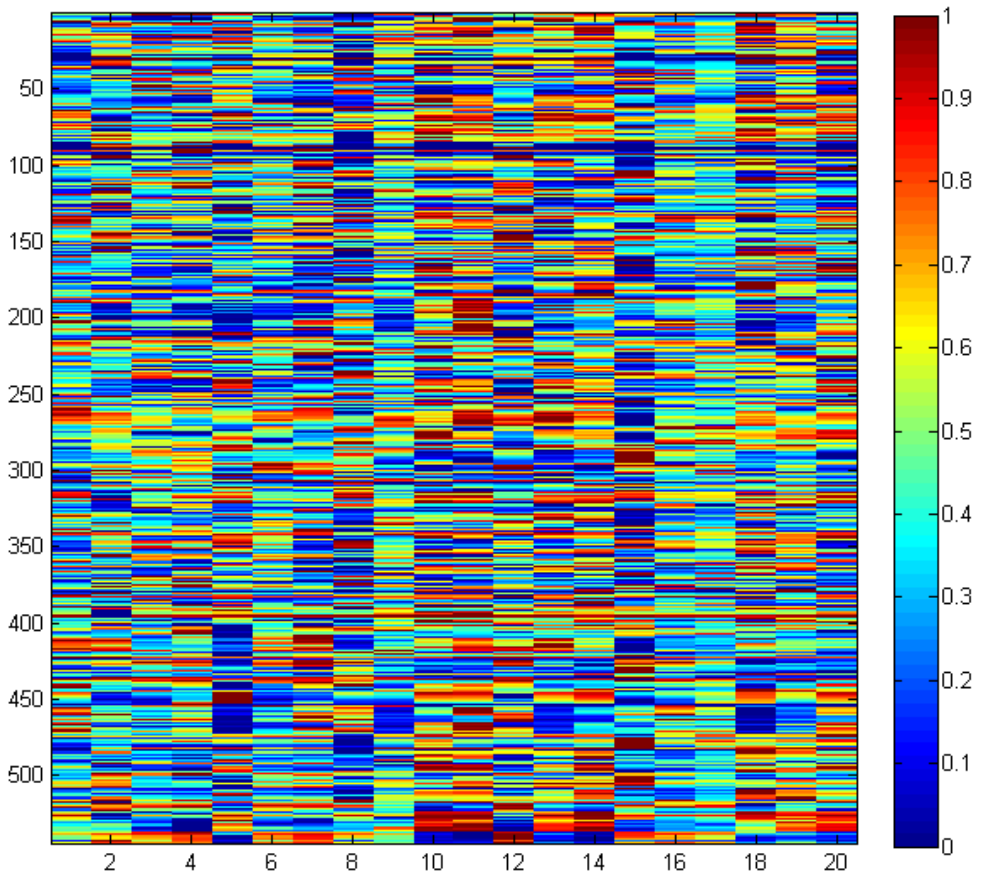


Figure 6.1 544 properties of amino acids from the AAindex database.

This figure shows the normalized color map values of the 544 amino acid physicochemical properties that were used for this study. Red is of high value and blue is of low value closer to zero. The varying colors represent the varying values for different amino acid physicochemical properties, which will help machine learning algorithms find good patterns in the sequence data when these values are encoded in the sequences. Good patterns can lead to higher accuracies for secondary structure prediction. These values are discussed under Section 6.2.1.

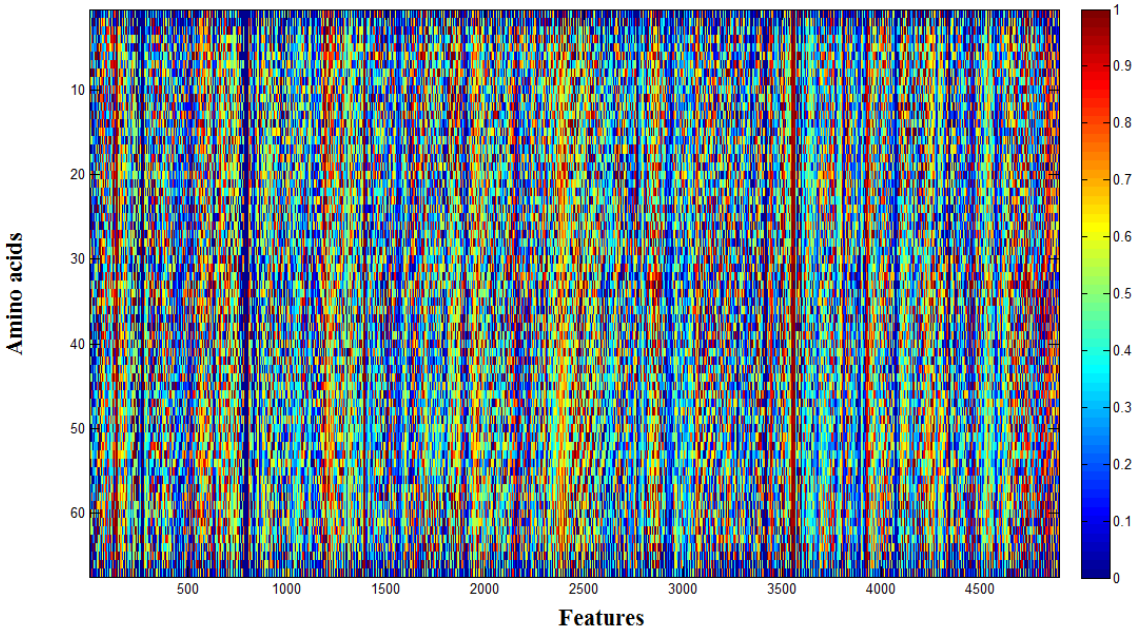


Figure 6.2 1ahb protein encoded with 4896 features derived from 544 amino acids properties.

This figure shows the normalized color map values of 1ahb protein encoded with 4896 features derived from 544 amino acid physicochemical properties that were used for this study. Each amino acid (each row on the vertial axis) is encoded with 4896 features. Red is of high value and blue is of low value closer to zero. The varying colors represent the varying values for different amino acid physicochemical properties, which will help machine learning algorithms find good patterns in the sequence data when these values are encoded in the sequences. Good patterns can lead to higher accuracies for secondary structure prediction. These features were reduced using **PCA** and **GA** as discussed under Section 6.3.

CHAPTER 7. IMPROVING SECONDARY STRUCTURE PREDICTION USING POSITION SPECIFIC RESIDUE PREFERENCES OF AMINO ACIDS

7.1 Abstract

In previous chapters, protein secondary structure predictions were obtained from knowledge-based potentials, using **FLOPRED** methodology. These predictions are improved using information extracted from the Position Specific Residue Preferences (**PSRP**) of amino acids present in a set of $1,860$ proteins. The influence of the lengths of secondary structures and the preferences of amino acids to appear at either end of the three secondary structures, α-helix, β-strand and coil are investigated. We find that **PSRP** can be employed to improve secondary structure prediction.

7.2 Secondary structure prediction with FLOPRED

Current methods of secondary structure prediction have accuracies slightly above 70% if only sequences information is used, while they achieve a prediction accuracy near 80% if the methods include multiple sequence alignments. The initial secondary structure predictions using **FLOPRED** are based on **ELM** classifications using advanced **PSO** algorithms to tune parameters such as the number of hidden neurons, weights and biases of the sigmoidal activation function. We are able to predict secondary structures with a training accuracy of 93.33% and a testing accuracy of 92.24% with a standard deviation of 0.48%,using a small set of 84 proteins as discussed in previous chapters. The low standard deviation and the small difference of less than 1% between the training and testing set shows good generalization performance. Here we investigate whether these accuracies can be improved further with the use

of information gleaned from position specific residue preferences and other information such as length of secondary structures and the frequency of occurrences of each secondary structure. We have conducted a preliminary study to where these values are used for voting on the class of secondary structure. The collective voting scores are used for the final secondary structure prediction as discussed in the results section.

7.3 Results obtained from FLOPRED

A set of small proteins (Dataset-84) has been selected from the **CB513** data set (Cuff and Barton, 2000) for secondary structure prediction, using the **FLOPRED** methodology. This set is a collection of small proteins with less than 125 residues each and a total of $7,500$ residues. The results obtained for secondary structure prediction using **FLOPRED** on this data set is much higher than those found in literature as discussed in section 4.4 on page 86.

Residue preferences of amino acids at the ends of secondary structures have been used for secondary structure prediction (Richardson and Barlow, 1999; Duan et al., 2008) . In order to improve **FLOPRED** classification results, we investigate here, the usefulness of *Position Specific Amino Acid Preferences* (**PSRP**) to appear at the ends of secondary structures. We will also look at the propensities of structures to appear within particular lengths and the number of occurrences of each secondary structure in a given set of proteins. We will investigate amino acid residue preferences for seven positions in an α-helix and five residue positions for β-strand and coil at the N-terminal end and the C-terminal end. Propensities for a given length and propensity for number of occurrences for secondary structures are also investigated to see if there are any possibilities for improving secondary structure prediction results.

All these structures are formed starting from the amino acid sequence. The folding itself is influenced by the several types of interactions between the residues which are determined by the physicochemical properties of the amino acids. We aim to investigate these features with respect to different secondary structures and their constituent amino acids. These amino acids might have position specific preferences to appear near the ends of secondary structures.

7.4 Initial studies to determine contribution of PSRP

The structure propensities for the training sequences and position specific amino acid propensities 7.1 for the amino acids in the full data set were initially determined, as listed below. The usefulness of the data, (as determined from the classification results) in contributing to increased prediction accuracy is discussed below. The different propensity values for various metrics are given in various figures.

- Structure propensities: as given in Figure 7.2 statistics on lengths of sequences between 20 and 30 amino acids do not contribute much to improve prediction accuracies, since they are not very significant as seen in the figure. Propensities for individual structures Table 7.1 such as α-helix, β-sheet and coil are found to be useful both as stand-alone parameters and when combined with other parameters and are used as a sort of rule set to determine membership to particular secondary structures.

- Amino Acid propensities : the occurrences of the 20 amino acid residues at ends of secondary structures as shown in Figure 7.1, is useful. As can be seen from the figure, each amino acid has a different propensity for the three secondary structures and these propensities are different for each of the 20 amino acids. AA propensities for particular residues at *specific positions* is also useful. Propensities for *particular residues* with *particular length* of a secondary structure (Helix, Sheet, Coil) in a *particular position* is a useful parameter in combination with *structure propensity for occurrence*. Propensities of amino acids to appear at the ends of secondary structures (Duan et al., 2008) is also an important contribution to determining secondary structure.

Many of these propensity values are used in the calculations and are given a vote in determining the secondary structure of the testing set. A sample rule set is shown in Table 7.1, where it shows that an alanine residue with an entry $1 - 0 - 0$ will be considered to be favored to be in an alpha helix rather than in a beta sheet or coil. Several of the **PSRP** values were similarly assigned votes so that they got a chance to vote on their choice of secondary structure, for the residue of interest.

Using the values in Table 7.1, an analysis was made on 699 test residues, to see if the **PSRP** values can help to increase prediction accuracy. It was found that if we disregard the **ELM** results and depend only on the votes cast by the **PSRP** values, we obtain only an extremely low accuracy rate of less than 50% for the first guess (first of three choices) and less than 13% (second of three choices). So, it was decided that it was prudent to keep the **ELM** results since they are almost 100% accurate for residues which are not at the ends of the secondary structures. So, **PSRP** values should be considered only for the residues at the ends of the structures, where the boundaries between secondary structure elements are located since the highest accuracies comes from the **ELM-PSO** method. If the **PSRP** results are combined with **ELM** then some of the correct entries are incorrectly classified and the overall accuracy percentage declines. At the same time it was found that the **PSRP** values do have some credibility since the *first Guess* was the correct class in some cases where the **ELM** gave the wrong classification (for some end residues). It seems that if we could find a way to somehow combine **ELM** and **PSRP** predictions, we would be able to improve the classification accuracy considerably over the **ELM-PSO** accuracies. But trying to increase the accuracies manually would be too arbitrary and too cumbersome when it comes to classification of thousands or even millions of residues. Hence we developed a methodology to model the **PSRP** values and incorporate them as part of the machine learning process to assess their contributions to secondary structure prediction, which is discussed next.

7.5 PSRP models for secondary structure prediction

Sequences from 1860 proteins with 511,648 residues (Duan et al., 2008) are used to calculate the **PSRP** values. The structure-changes from one structure to another, either at the beginning of a structure or at the end of a structure, can be expressed in terms of five residues (5%-mers) starting from two residues just before the start (or just after the end) of a structure and three residues which are part at the beginning of the secondary structure element itself (or at the end). For the first set of patterns, which are pattern changes at the beginning of secondary structures, the first two letters are replaced by the six combinations of **H, E C**

followed by patterns **HHH, EEE or CCC**. This yields a total of 27 patterns ($3^2 * 3$). Similarly in the next set the last two letters are replaced by the six combinations of **H, E C**, preceded by the three patterns **HHH, EEE or CCC**. Pattern-changes like **HHH HH** are also included for completion. The middle residue is the residue of interest for which we seek secondary structure assignment. There are a total of $27 + 27 = 54$ possible patterns of 5-mers. Only 45 are used in this study. The remaining 9 patterns which were not investigated in this study will be considered in future. Of the 45 patterns, only 13 cases capture most of the occurrences of which *only* 8 *represent over* 4% *of the residues* as seen in Figure 7.5. Only these most frequent 8 patterns are used in our analysis. The counts of the remaining patterns are relatively insignificant and have not been included. For example, the pattern *HHCCC* is a boundary between an α-helix structure and a coil. Besides these boundaries we also consider non-boundary, continuing structures such as *HHHHH, EEEEE, CCCCC*. 45 of these models consisting of $359,579$ residues were considered, and a list of these is shown in Table 7.2. This table shows the data for all the 45 cases for three secondary structures.

Of these 45 types, only 13 share most of the residues as shown in Figure 7.5 and Table 7.3. The number of residues in each of the three secondary structures is given in Figure 7.3 and the number of residues in each of the 13 cases are given in Table 7.3. Of these 13 models, only 8 share over 4% of the residues (as seen in the data and figure) and these are the only ones used in our analysis.

The dataset of 1860 sequences were scored for each of the 45 patterns and the number of occurrences of each of the 20 amino acids in the five positions in the patterns are stored. These statistics are used to encode the sequence features and are used during secondary structure prediction. The **ELM-PSO** algorithm will train on some of the sequences encoded with these features and then predict the secondary structure of the central residue in a 5-mer pattern. The values calculated for the pattern *HHHHH* are given in Table 7.4 and Figure 7.6, which shows the propensities for each of the 20 amino acids for each of the five positions. Here the residue of interest is the third or the middle residue. If the pattern is *HHEEE* then the residue of interest is **E**. This figure shows a rich variety of colors where red indicates a high value or

higher propensity to occur at one of the five positions compared to blue which indicates lower propensity. Such tables were built for all these cases. For each of the 8 cases, the sequences are encoded with the values in these tables. Then these features are used instead of the traditional orthogonal or **PSSM** values.

7.6 Results and discussion

The **ELM-PSO** is run on the sequences encoded with the **CABS** potential data, as discussed in earlier chapters. The ELM-PSO algorithm used for these runs is the general PSO algorithm and *not the advanced PSO algorithms* which were used in **FLOPRED**. Hence the accuracies obtained for secondary structure prediction for these runs is nearer to 79% and not higher as seen in later studies using **FLOPRED**. Then the same sequences are encoded with the feature values derived for each of the models. 8 different classifications are done and the accuracy for secondary structure classification is given in Figure 7.7. The patterns for which models were built are shown in the figure. Depending on the pattern, each of these data will have a different number of positive and negative patterns since all patterns do not occur uniformly in the data set. The tables show that the **ELM-PSO** results of secondary structure classification for the sequences coded with just the **CABS** potentials data is on an average 79% while the accuracies for secondary structure classification using only the **PSRP** feature values vary between 25% for Model M4, corresponding to the *HHHCC* pattern and 71% for the model M2 corresponding to *EEEEE* pattern. In these models the middle residue is the one for which secondary structure is predicted. The combined accuracy for these two models ranges between 83% and 93% but some of these accuracies will be overlapping where the predictions for the **ELM-PSO** and those for **PSRP** are both correct. So, the actual accuracies can range between 78% and 93% depending on which model we choose. The last line in the table gives the contribution that the **PSRP** values could potentially make to increase secondary structure prediction accuracies and these values range between 6% and 15%. These are results of preliminary runs on about 1000 residues. The advanced PSO optimizations of **FLOPRED** have not yet been performed on any of these data. Using the **PSRP** values derived from

these analyses, we can make a more thorough study using larger and newer protein data sets to check whether the **PSRP** can indeed make useful contributions to increases in secondary structure prediction accuracies.

7.7 Conclusions and future work

Feature values for different structure-changes patterns found in protein sequences have been derived. Amino acid counts found in these positions were used to calculate these feature values, which are the propensities of each amino acid to occur in particular positions at the beginning and end of the three secondary structures. Secondary structure classifications for the same set of sequences encoded with CABS potentials data and the **PSRP** features were obtained using **ELM-PSO** algorithm. The results were compared and combined to estimate the minimum and maximum gains that can be obtained in secondary structure prediction accuracies using **PSRP** feature values. Future work in this area would use **PSRP** values on larger data set and optimize the classification results using the more advanced **FLOPRED** algorithm for classification.

It is to be noted that only a trial run with many combinations of the **PSRP** values has been carried out in this project. A more elaborate scheme that would include the **PSRP** values as part of the **ELM** data itself needs to be made before any conclusions can be drawn. This will prevent any subjective and biased rules being imposed on the classifications. We can also use only **PSRP** values to isolate their contribution to the classification of secondary structures.

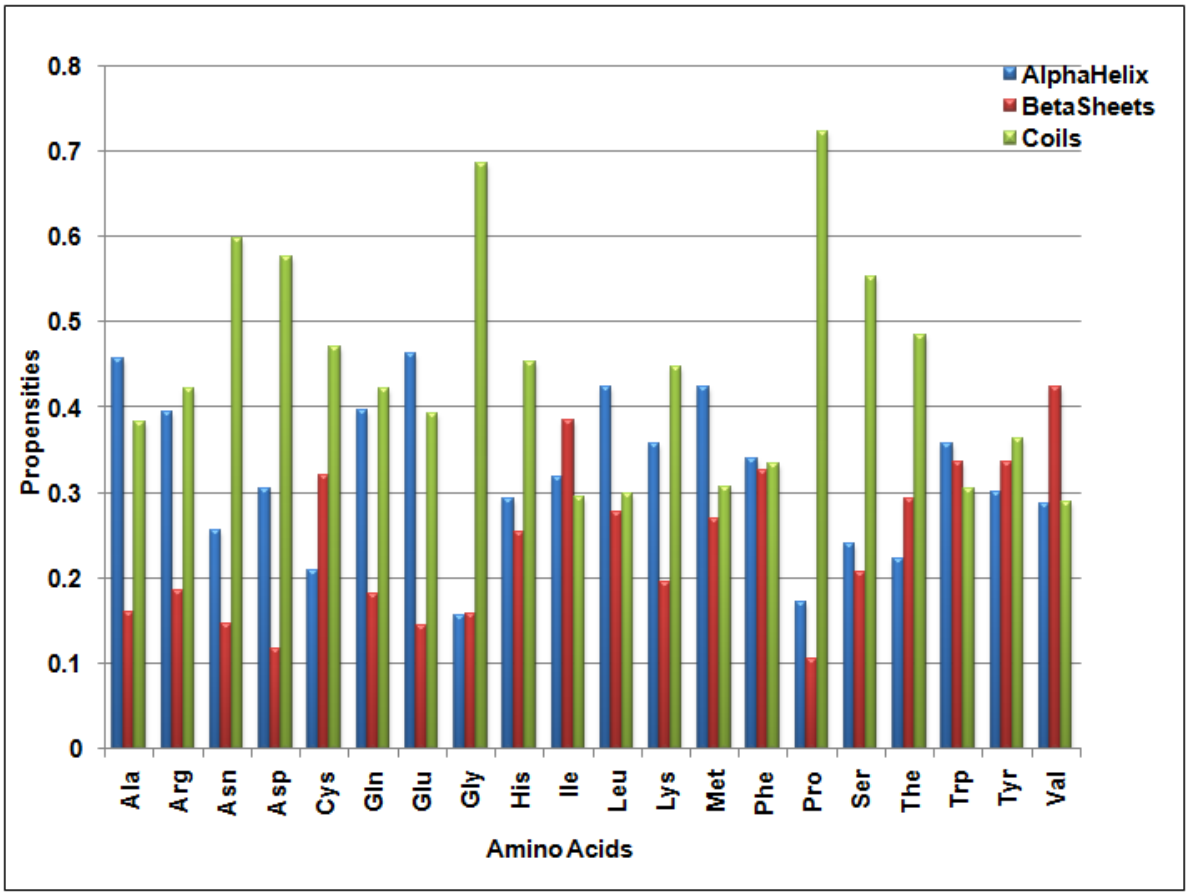


Figure 7.1 The propensities of the 20 amino acids in secondary structures

Data for this graph is derived from 1860 protein sequences (Duan et al., 2008). The percentage of each of the 20 amino acids , in each of the three secondary structures present in this data set, is shown in this figure. We observe that some of the amino acids such as alanine, glutamic acid, leucine, methionine and tryptophan prefer to be present in α-helices while others such as isoleucine and valine prefer β-strands while the remaining residues prefer coil. This figure clearly illustrates that preferences of the various amino acids for secondary structures is non-uniform.

Table 7.1 Propensities of 20 amino acids to appear at the ends of secondary structures.

Data for this graph is derived from 1860 proteins (Duan et al., 2008). This figure shows the number, letter codes and names of 20 amino acids. The number 1 indicates a preference and a 0 indicates absence of preference (relative to the structure having greater preference). Some of the amino acids such as alanine, glutamic acid, phenylalanine, leucine, methionine and tryptophan prefer to appear at the ends of α-helices more than other secondary structures while isoleucine and valine prefer to appear at the ends of β-strands and the remaining residues prefer coil. These values were calculated based on the statistics of the content of the amino acids at the ends of secondary structures in the given dataset.

#	**Letter code**	**3-letter code**	**Amino acid name**	$\alpha - helix$	$\beta - sheet$	**Coil**
1	A	Ala	Alanine	1	0	0
2	C	Cys	Cysteine	0	0	1
3	D	Asp	Aspartic Acid	0	0	1
4	E	Glu	Glutamic Acid	1	0	0
5	F	Phe	Phenylalanine	1	0	0
6	G	Gly	Glycine	0	0	1
7	H	His	Histidine	0	0	1
8	I	Ile	Isoleucine	0	1	0
9	K	Lys	Lysine	0	0	1
10	L	Leu	Leucine	1	0	0
11	M	Met	Methionine	1	0	0
12	N	Asn	Asparagine	0	0	1
13	P	Pro	Proline	0	0	1
14	Q	Gln	Glutamine	0	0	1
15	R	Arg	Arginine	0	0	1
16	S	Ser	Serine	0	0	1
17	T	Thr	Threonine	0	0	1
18	V	Val	Valine	0	1	0
19	W	Trp	Tryptophan	1	0	0
20	Y	Tyr	Tyrosine	0	0	1

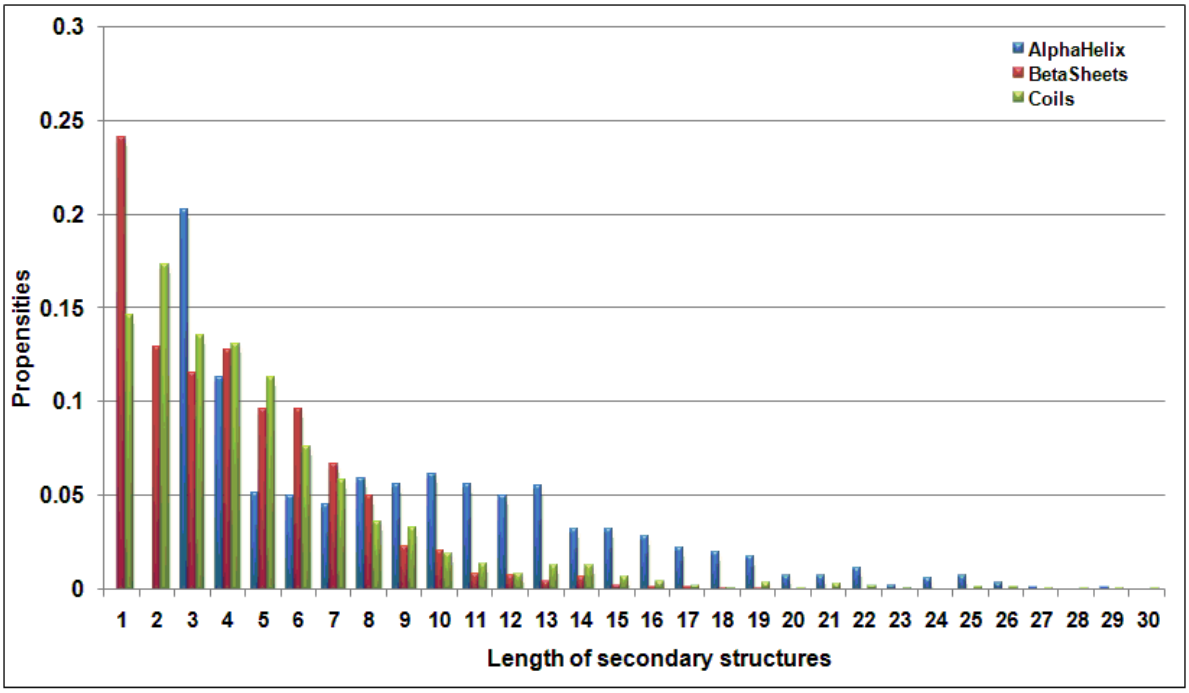


Figure 7.2 Length distributions of the 20 amino acids in secondary structures

Data for this graph is derived from 1860 proteins (Duan et al., 2008). The counts of the lengths of secondary structures present in this dataset for each of the three secondary structures, is shown in this figure as a percentage. This figure shows that most of the secondary structures found in this dataset have lengths ranging from single residues to around 10 residues (although helices start at 3 residues, which might include half turns). α-helices tend to be longer than other secondary structures and coils are also longer than β-strand. The length propensities for sequences of length between 20 and 30 are not found to contribute significantly to secondary structure prediction, possibly because not enough sequences of those lengths are available.

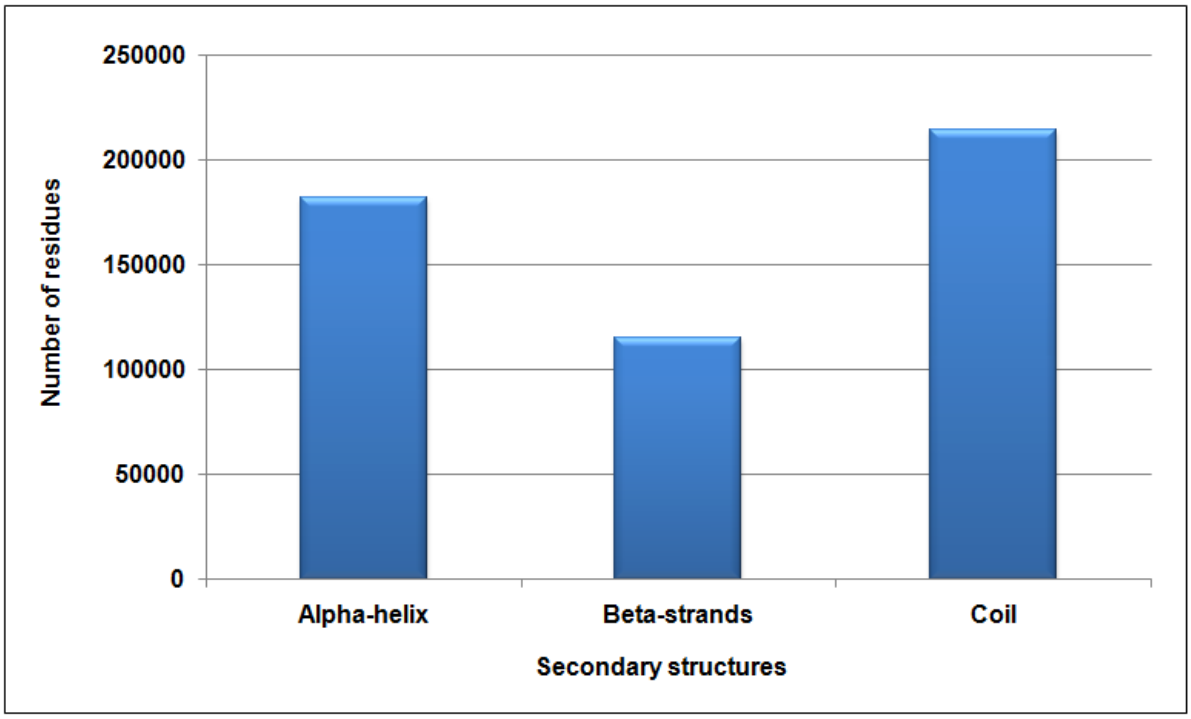


Figure 7.3 Secondary structure counts in **PSRP** analysis

This figure shows the number of residues in each of the three secondary structures used for the **PSRP** analysis. The data is derived from the a data set of 1860 proteins (Duan et al., 2008).

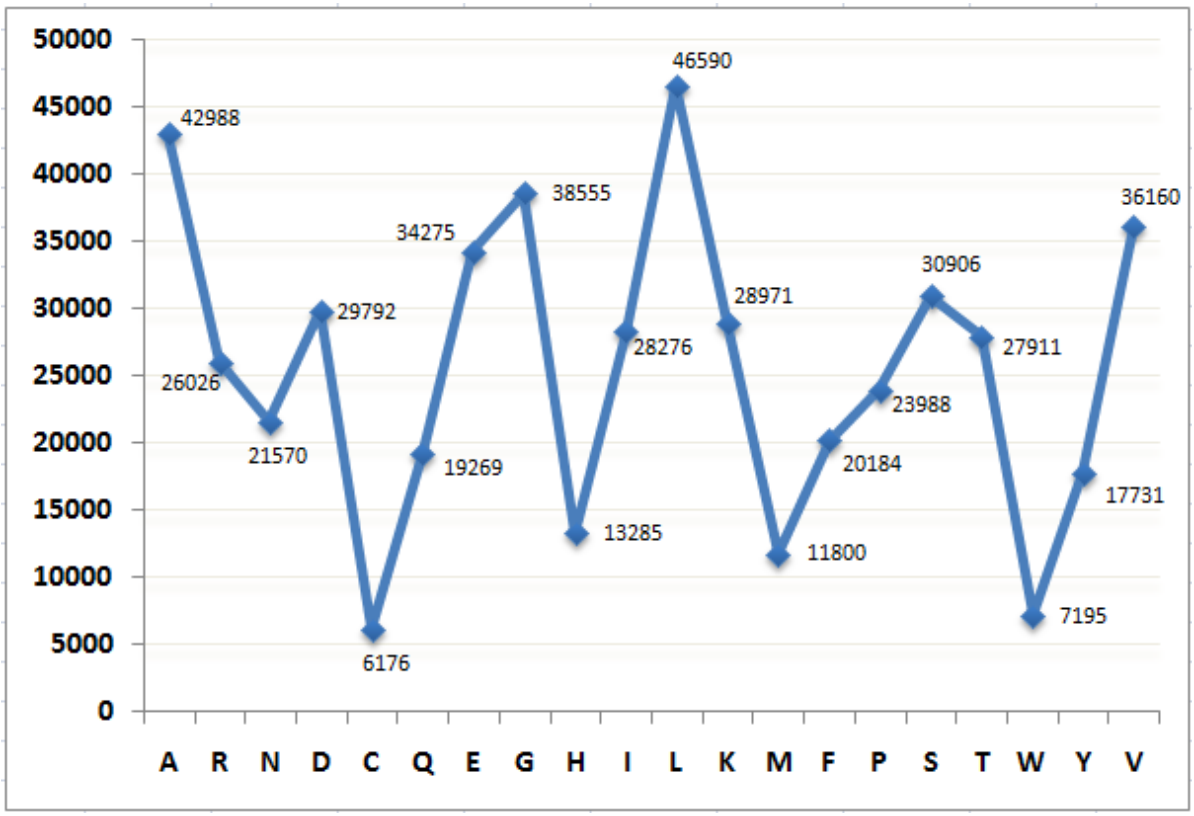


Figure 7.4 Amino acid counts for the full data set in **PSRP** analysis

The data is derived from the a data set of 1860 proteins (Duan et al., 2008). This figure shows the number of residues for each of the 20 amino acids in the full dataset used for the **PSRP** analysis

Table 7.2 **PSRP** models - 5-mers of patterns

The data for this table is extracted from 1860 protein sequences (Duan et al., 2008). This figure shows the counts for 45 structure boundaries between different secondary structures. For the first set of patterns, which are pattern changes at the beginning of secondary structures, the first two letters are replaced by the six combinations of **H, E C** followed by patterns **HHH, EEE or CCC**. This yields a total of 27 patterns ($3^2 * 3$). Similarly in the next set the last two letters are replaced by the six combinations of **H, E C**, preceded by the three patterns **HHH, EEE or CCC**. Pattern-changes like HHH HH are also included for completion. The middle residue is the residue of interest for which we seek secondary structure assignment. There are a total of $27 + 27 = 54$ possible patterns of 5-mers. Only 45 are used in this study. The remaining 9 patterns which were not investigated in this study will be considered in future. Of the 45 patterns, only 13 cases capture most of the occurrences of which *only* 8 *represent over* 4% *of the residues* as seen in Figure 7.5. Only these most frequent 8 patterns are used in our analysis. The counts of the remaining patterns are relatively insignificant and have not been included. *struct* gives the secondary structures of which the central residue is the class of the pattern, count gives the number of patterns of this type that is found in the dataset, % gives the fraction this pattern found in the full dataset.

ID	Pattern	Struct	Cls	Count	%	ID	Pattern	Struct	Cls	Count	%
1	HHHHH	11111	1	108386	30	24	EEECH	22231	2	1064	0
2	HEHHH	12111	1	14	0	25	EEECE	22232	2	489	0
3	EEHHH	22111	1	1096	0	26	EEECC	22233	2	14915	4
4	CEHHH	32111	1	395	0	27	CCCCC	33333	3	77935	22
5	HCHHH	13111	1	1415	0	28	HHCCC	11333	3	13261	4
6	ECHHH	23111	1	1725	0	29	EHCCC	21333	3	0	0
7	CCHHH	33111	1	14687	4	30	CHCCC	31333	3	0	0
8	HHHEH	11121	1	14	0	31	HECCC	12333	3	178	0
9	HHHEE	11122	1	616	0	32	EECCC	22333	3	12918	4
10	HHHEC	11123	1	251	0	33	CECCC	32333	3	2529	1
11	HHHCH	11131	1	1415	0	34	CCCHH	33311	3	11678	3
12	HHHCE	11132	1	982	0	35	CCCHE	33312	3	0	0
13	HHHCC	11133	1	15974	4	36	CCCHC	33313	3	0	0
14	EEEEE	22222	2	37295	10	37	CCCEH	33321	3	280	0
15	HHEEE	11222	2	531	0	38	CCCEE	33322	3	14290	4
16	EHEEE	21222	2	0	0	39	CCCEC	33323	3	2765	1
17	CHEEE	31222	2	0	0	40	HHEHH	11211	2	14	0
18	HCEEE	13222	2	529	0	41	HHCHH	11311	3	1415	0
19	ECEEE	23222	2	476	0	42	EEHEE	22122	1	0	0
20	CCEEE	33222	2	15835	4	43	EECEE	22322	3	344	0
21	EEEHH	22211	2	907	0	44	CCHCC	33133	1	0	0
22	EEEHE	22212	2	0	0	45	CCECC	33233	2	2961	1
23	EEEHC	22213	2	0	0		Total			359579	1

Table 7.3 T

his data is derived from a data set of 1860 proteins (Duan et al., 2008). This histogram shows the number of residues for each of the 13 models used in the **PSRP** analysis. Some patterns such as HHHEE, HHHCC, EEECC, CCCHH, CCCEE have few occurrences (relative to the total) and are combined with other models during our studies. Some patterns present in negligible quantities are not used in our analysis. The data for this figure is given in Table 7.3 and is further discussed in Section 7.5

Model Number	**Pattern**	**Number of Residues**	**%**
1	HHHHH	108386	0.32
2	CCHHH	14687	0.04
3	HHHEE	616	0
4	HHHCC	15974	0.05
5	EEEEE	37295	0.11
6	HHEEE	531	0
7	CCEEE	15835	0.05
8	EEECC	14915	0.04
9	CCCCC	77935	0.23
10	HHCCC	13261	0.04
11	EECCC	12918	0.04
12	CCCHH	11678	0.04
13	CCCEE	14290	0.04
	Total	338321	

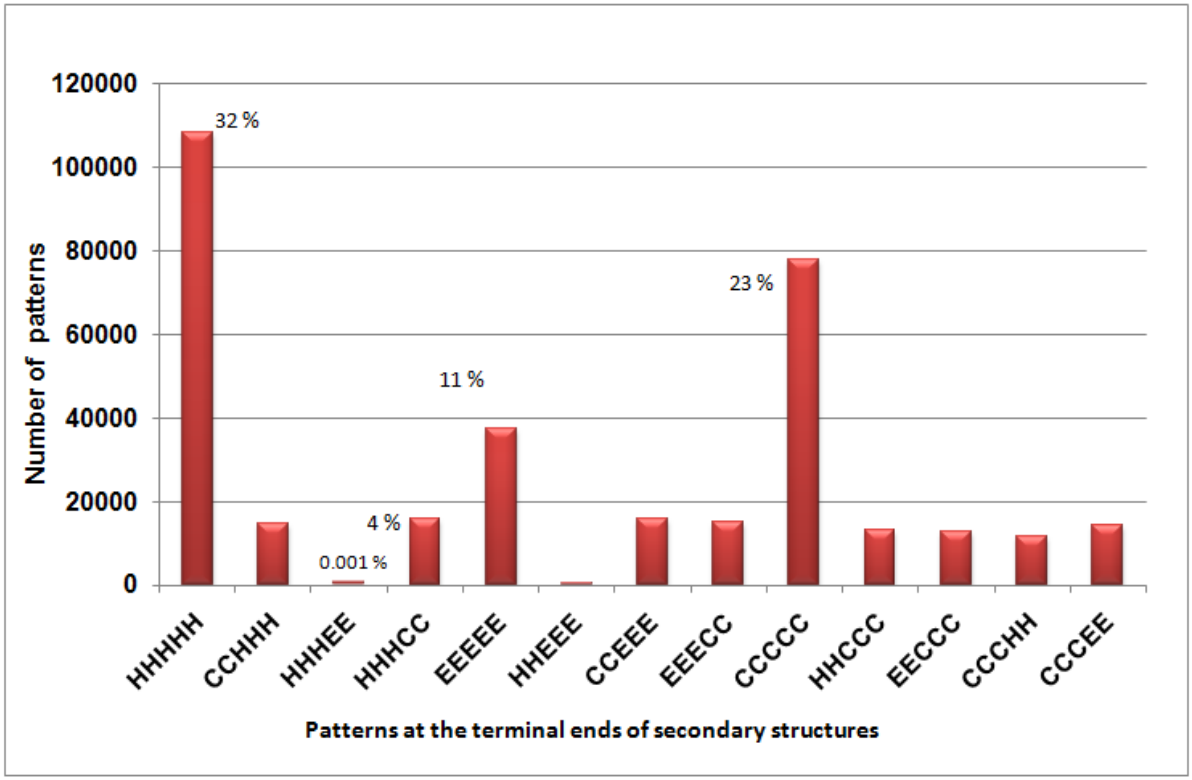


Figure 7.5 Amino acid counts for 13 models in **PSRP** analysis.

The data is derived from the a data set of 1860 proteins (Duan et al., 2008). This figure shows the number of residues in each of the 13 models used for the **PSRP** analysis. Some patterns such as HHHEE, HHHCC, EEECC,CCCHH, CCCEE which are few in number (relative to number of occurrences of other patterns) were combined with other models during our studies. Some patterns which were present in negligible quantities are not used in our analysis. This data is given in Table 7.3 and is further discussed in Section 7.5

Table 7.4 Propensities of the 20 amino acids to appear in the HHHHH pattern

The data is derived from the data set of 1860 proteins (Duan et al., 2008) and represents the propensities of amino acids at the five different positions in the *HHHHH* pattern. Sequences are coded with these features (for a window size of 9) for secondary structure prediction. Similar tables are used for the other 12 models. A model is built with the set of protein sequences encoded with the features in this table that are used for secondary structure prediction of the central residue in each pattern. A graphical view of these features is given in Figure 7.6, where the larger values are colored red and the blue ones are small values close to zero.

Amino Acid	**1**	**2**	**3**	**4**	**5**
A	0.12	0.12	0.13	0.13	0.13
R	0.06	0.06	0.06	0.07	0.07
N	0.03	0.03	0.03	0.03	0.03
D	0.05	0.05	0.04	0.04	0.04
C	0.01	0.01	0.01	0.01	0.01
Q	0.05	0.05	0.05	0.05	0.05
E	0.10	0.09	0.08	0.08	0.08
G	0.04	0.04	0.03	0.03	0.03
H	0.02	0.02	0.02	0.02	0.02
I	0.06	0.06	0.07	0.07	0.06
L	0.11	0.12	0.13	0.14	0.13
K	0.06	0.06	0.06	0.07	0.07
M	0.03	0.03	0.03	0.03	0.03
F	0.04	0.04	0.04	0.04	0.04
P	0.03	0.01	0.01	0.01	0.01
S	0.04	0.04	0.04	0.04	0.04
T	0.04	0.04	0.04	0.04	0.04
W	0.02	0.02	0.02	0.02	0.01
Y	0.03	0.03	0.04	0.04	0.04
V	0.07	0.07	0.07	0.07	0.06

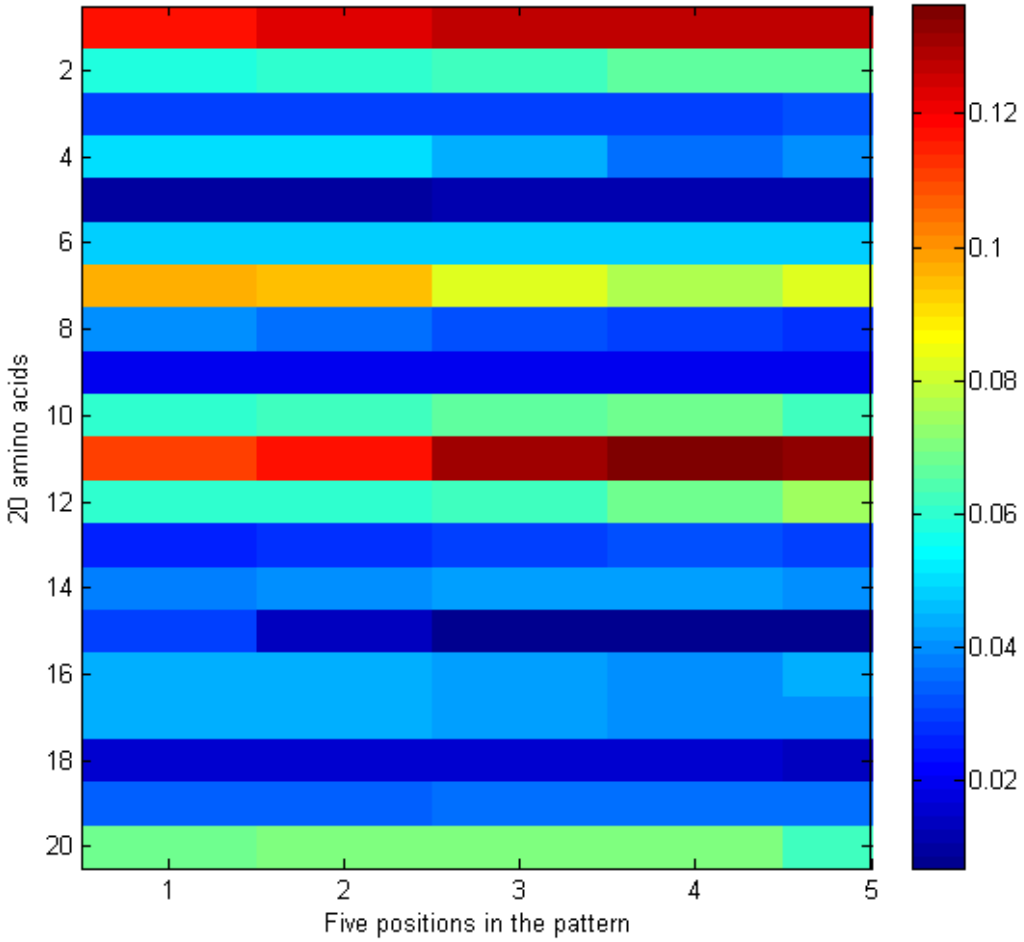
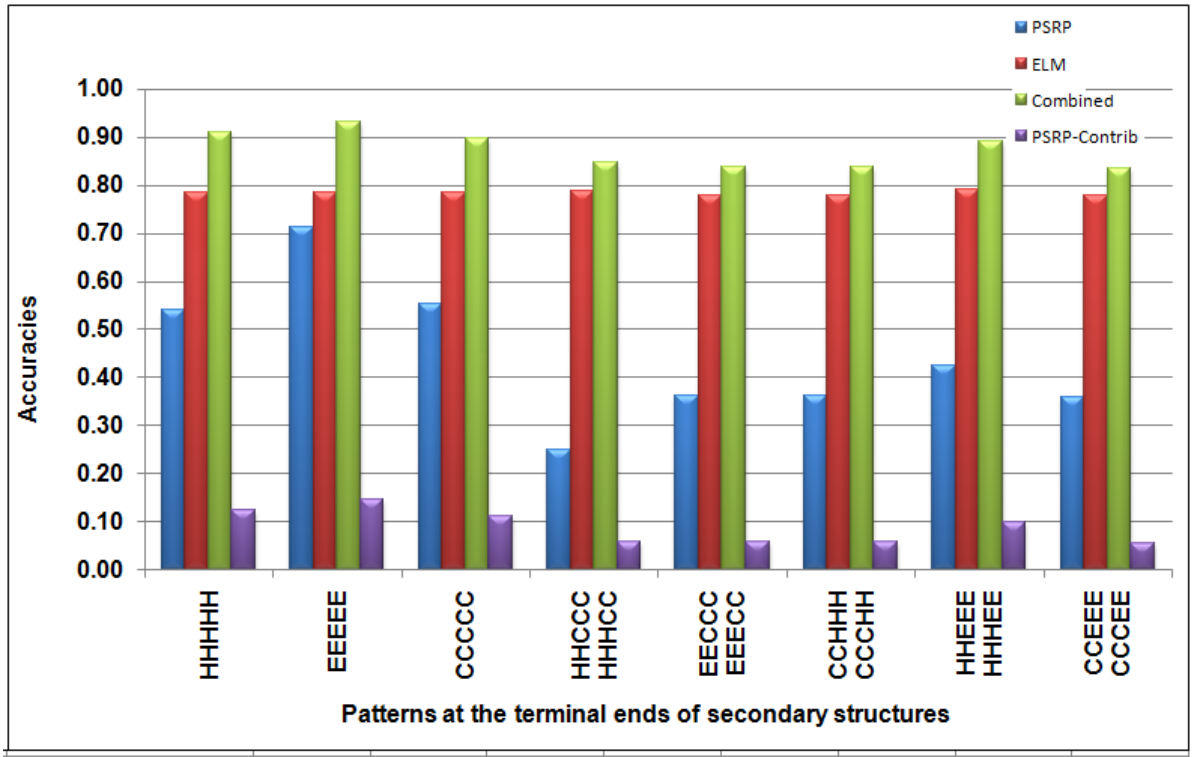


Figure 7.6 Color map of feature values for the HHHHH pattern in **PSRP** analysis

The data is derived from the a data set of 1860 proteins (Duan et al., 2008). This table shows the color map of the propensities of all amino acids in the 'HHHHH' pattern. Sequences are coded with these features (for a window size of 9) for secondary structure prediction. Similar tables are used for the other 12 models. This figure shows that the 20 amino acids have different propensities to appear at different positions in the HHHHH pattern. Some of the residues have very high propensity (red) to appear in the H secondary structure while others have very low propensities (blue). This information could be mined to improve secondary structure prediction. The data values for this color map is given in Table 7.4. These propensities are discussed further in Section 7.5



PSRP	0.54	0.71	0.55	0.25	0.36	0.36	0.42	0.36
ELM	0.79	0.79	0.79	0.79	0.78	0.78	0.79	0.78
Combined	0.91	0.93	0.90	0.85	0.84	0.84	0.89	0.83
PSRP-Contrib	0.12	0.15	0.11	0.06	0.06	0.06	0.10	0.06

Figure 7.7 Classification accuracy for the 3 secondary structures

The data for the patterns is derived from the 1860 protein set (Duan et al., 2008). The features are encoded in the CB513 protein sequences (Cuff and Barton, 2000) and used for secondary structure prediction. This figure gives the classification accuracies for the three secondary structures for different patterns at the ends of secondary structures. Accuracies for some patterns, as illustrated in the figure, are combined with others (if the total count of a pattern is small in comparison to occurrences of other patterns). Table 7.2 gives the counts of each pattern that occurs in the dataset. The accuracies using only PSRP features gives accuracies between 26% and 71%. Secondary structure accuracies using knowledge-based potentials and general PSO are 78% to 79%, as shown in the figure. The combined accuracies, where both models give correct classifications is between 84% and 93%. But, these including overlapping accuracies, where both models (ELM and PSRP) are correct. The contribution from PSRP is expected to be between 6% and 12%. These results are discussed further in Section 7.6

CHAPTER 8. Conclusions and future studies - Part II

8.1 Secondary structure prediction using knowledge-based potentials

Improved protein secondary structure prediction methods using machine learning and optimization are implemented. Data is generated using **CATH** library (Orengo et al., 1997; Cuff et al., 2008) structures and **CABS** (Kolinski, 2004) force field, to encode long and short range interaction information that is present in the **CB513** (Cuff and Barton, 2000) protein sequences. Sequences which shared more than 20% pair-wise sequence similarity or any structure similarity with the **CATH** structure templates are removed from the study. A modified form of neural network called Extreme Learning Machine **(ELM)** (Huang et al., 2006) is used to develop a multi-class algorithm for secondary structure classification of three secondary structure types; α-helix, β-strand and coil. Use of advanced Particle Swarm Optimization (**PSO**) (Kennedy and Eberhart, 1995) techniques leads to gains in processing time and improvements in the quality of predictions, as more of the machine learning parameters are included in the PSO optimization.

Two sets of data from the **CB513** dataset are used for our studies, dataset-84 which has 84 proteins and dataset-415 which has 415 proteins. Initially, the **ELM-PSO** algorithm yields an average accuracy of 79% using knowledge-based potentials data (Saraswathi et al., 2011), where a general **PSO** is used to tune the weights and the biases of **ELM** (Saraswathi et al., 2010b). Later on, an improved **PSO** algorithm with advanced capabilities (Fernández-Martínez and García-Gonzalo, 2008, 2009, 2010) has been used and other parameters such as the number of hidden neurons, lambda values and other **PSO** parameters are included in the algorithm, which results in the improved **FLOPRED** algorithm that is used in many of our applications. The improved techniques lead to gains in accuracies of between 2% and 3% at

each stage of our studies. Our final Q_3 accuracies for dataset-84 are 93.33% for training and 92.24% for testing, with a standard deviation of 0.48%. A testing accuracy of 86.5% with a standard deviation of 1.38% is obtained on dataset-415. A comparative study against similar work in the literature on the **CB513** dataset of proteins sequences indicates that our results are better by almost 6% for dataset-84 and is by less than only 0.44% for dataset-415. In addition, our algorithm is much simpler, faster and needs fewer resources to achieve these results. The only drawback of our algorithm is in the time required for the computationally intensive data generation for training and test sequences.

Our future work will aim to improve prediction accuracies by utilizing larger sets of Sequences for training and testing.

8.2 An amino acid perspective of secondary structure prediction

Our secondary structure prediction results obtained by using **FLOPRED** were analyzed with respect to their amino acid content. We found that many types of amino acids contributed to the high accuracies obtained in our results although they were present in lower quantities, compared to other amino acids, contradicting the usual perception that increased representation in the data used for training set would lead to higher accuracies.

Future studies will involve further investigation of secondary structure prediction results at the amino acid level to confirm our findings. We would like to investigate a multi-class algorithm that would classify secondary structures on the basis of the membership of amino acids in secondary structures. This will be a multi-class problem involving classification of 60 classes, where there will be 20 classes for each amino acid in each of the three secondary structures.

8.3 Secondary structure prediction using physicochemical properties of amino acids

544 physicochemical properties of amino acids from the **AAindex** database AAindex1 (Kawashima et al., 1999) database are used in an initial study, on a small set of proteins, to

determine the contribution of physicochemical properties to secondary structure predictions. A Genetic Algorithm and Principal Component Analysis are used to reduce this large number of features. The results of this study show results as good as those that use orthogonal data representation for secondary structure predictions.

Future studies will try to improve the current prediction accuracies in addition to identifying the most important physicochemical properties that lend themselves to improved secondary structure predictions.

8.4 Position specific residue preferences of amino acids at ends of secondary structures

Amino acid occurrences at five positions near the two ends of secondary structure segments are used as features to encode protein sequences. **ELM-PSO** is used to determine the secondary structure of these sequences. Initial studies indicate that the position specific residue preferences of amino acids may contribute to some increased secondary structure prediction accuracies.

Future studies will implement the **FLOPRED** algorithm on this data and will attempt to determine the contribution of position specific residue preferences of amino acids to secondary structure prediction.

In summary, several secondary structure prediction schemes are implemented in this study, with some of them showing good results while others show promise. We hope to improve these results in our future studies.

PART III

RELATIVE SOLVENT ACCESSIBILITY PREDICTION

CHAPTER 9. AN EXTREME LEARNING MACHINE CLASSIFIER FOR PREDICTION OF RELATIVE SOLVENT ACCESSIBILITY IN PROTEINS

A paper published in the Proceedings of IJCCI/ICNC 2010[1]

Saras Saraswathi [2,3], Robert L. Jernigan[2] and Andrzej Kloczkowski[2,4]

Keywords

Relative Solvent Accessibility, Support Vector Machine, Neural Network, Extreme Learning Machine, prediction.

Abstract

A neural network based method called Sparse-Extreme Learning Machine (**S-ELM**) has been used for predicting Relative Solvent Accessibility (**RSA**) in proteins. We have shown that multiple-fold gains in speed can be achieved by the proposed **S-ELM** algorithm compared to using Support Vector Machines (**SVM**) for **RSA** prediction. Classification accuracies obtained by the **S-ELM** algorithm are comparable to those in literature. This study indicates that using **S-ELM** would give a distinct advantage in terms of processing speed and performance for **RSA** prediction in proteins.

[1]Reprinted with permission of IJCCI/ICNC, 2010, ISBN 978-989-8425-32-4, pp. 364-369.

[2]Graduate student and Professors, respectively, Department of Biochemistry, Biophysics, and Molecular Biology, L .H. Baker Center for Bioinformatics and Biological Statistics, Iowa State University

[3]Primary researcher and author

[4]Author for correspondence

9.1 Introduction

Proteins perform a variety of important biological functions that are imperative to the wellbeing of all living things. Various factors determine protein functions, such as, its native structure, the information coded in its constituent amino acid sequences, its reactions to the surrounding solvent environment and the Relative Solvent Accessibility (**RSA**) values of its residues. Evaluating **RSA** values will help to gain an insight into the structure and function of a protein. Protein structures and other related values such as **RSA** can be experimentally determined by using **NMR** spectroscopy or X-Ray crystallography. But these methods can be expensive in terms of cost, time and other factors. There is an urgent need to process large amounts of data (spawned by advances in biotechnology) accurately and speedily in order to decipher the information buried in biological data, since it is impractical to do it manually. Computational methods such as machine learning algorithms provide an alternate way by which we can study this data in a cost and time efficient manner. Still, accuracies and processing efficiencies in existing methods are inadequate and there is a need for improvement. This study endeavors to attain a large gain in processing efficiencies.

RSA prediction has contributed to the study of protein functions in many applications. **RSA** can be used to determine protein hydration properties (Ooi et al., 1987), temperature sensitive residues can be identified and targeted for mutagenesis or it can be used to determine residues in contact (Shen and Vihinen, 2003). **RSA** has been used to improve secondary structure prediction (Adamczak et al., 2004) and for fold recognition and protein domain (DOMpro) prediction (Cheng and Baldi, 2006). **RSA** values can be used to gauge the degree of solvent exposure of segments of globular proteins (Carugo, 2000), to find residues with potential structural or functional (ConSeq) importance (Berezin et al., 2004), to help in the rationale design of antibodies and other proteins to improve binding affinities (David et al., 2007). In general **RSA** values can help to achieve cost and time efficiencies in drug discovery processes and help to gain a better understanding of biological processes. Probability profiles have been used (Gianese et al., 2003) to predict **RSA** values from single sequence and Multiple Sequence Alignment (**MSA**) data. **RSA** values can also be estimated from an atomic perspec-

tive (Singh et al., 2006). Homologous structural information can be used (Pollastri et al., 2007) to improve **RSA** prediction. In addition, tertiary structure predictions are increasingly being augmented and improved with information derived from secondary structures and **RSA** values. It has also been shown (Zarei et al., 2007) that pairs of residues can influence **RSA** prediction accuracy. Knowledge-based tools which use machine learning techniques and statistical theory can be valuable in predicting **RSA**, especially in the absence of evolutionary information or when sequences are not well conserved. Neural Networks have popularly been used for **RSA** prediction (Ahmad and Gromiha, 2002; Adamczak et al., 2004; Cheng et al., 2006). **RSA** values have been used (Pollastri et al., 2002) for scoring remote homology searches and for modeling protein folding and structure using a bidirectional recurrent neural network (ACCpro). Other methods used for **RSA** prediction include Information Theory (Naderi-Manesh et al., 2001), Multiple Linear Regression (Pollastri et al., 2002; Wagner et al., 2005), Support Vector Machines (Nguyen and Rajapakse, 2005) and fuzzy K-nearest neighbor algorithm (Sim et al., 2005). **SVM**psi and long range interactions have also been used (Kim and Park, 2004) to improve **RSA** accuracy. In order to compare their capabilities for **RSA** prediction, five different methods; Decision Tree (DT), Support Vector Machine (**SVM**), Bayesian Statistics (BS), Neural Network (NN) and Multiple Linear Regression (MLR) were applied to the same data set (Chen et al., 2004). The authors conclude that NN and **SVM** were among the best methods that were suitable for **RSA** prediction. More recently, sequence and structural information (Bondugula and Xu, 2008) were combined to estimate **RSA** values (MUPRED). A reliability Z-score has been developed (Petersen et al., 2009) to measure the degree of trust that can be related to individual predictions of **RSA**. A two-step approach has been developed (Meshkin and Ghafuri, 2010) using feature selection on physicochemical properties of residues and Support Vector Regression (SVR) to predict **RSA**. We propose to use a new fairly new method called Sparse Extreme Learning Machine (**S-ELM**), based on neural networks, which is capable of extreme speeds compared to traditional neural networks while maintaining current classification accuracies.

This paper is organized as follows. Section 2 on methods and data briefly discusses the

S-ELM algorithm and characteristics of the **RSA** data. Section 3 discusses the results of this study with performance comparisons with **SVM** and **NETASA** methods followed by conclusions in Section 4.

Methods and data

9.1.1 Extreme Learning Machine

Single Layer Feed-forward Networks (**SLFN**), with a hidden layer and an activation function possess an inherent structure suitable for mapping complex characteristics, learning and optimization. These networks have applications in bioinformatics for solving various problems like pattern classification and recognition, structure prediction and data mining. The free parameters of the network are learned from given training samples using gradient descent algorithms that are relatively slow and have many issues in error convergence. It has been proved theoretically (Huang et al., 2006) that a modified **SLFN** model called an Extreme Learning Machine (ELM), can provide good generalization performance and overcome some of the problems associated with traditional NNs such as stopping criterion, learning rate, number of epochs and local minima. **ELM** has good generalization capabilities and capacity to learn extremely fast. The input weights are chosen randomly but the output weights are calculated analytically using a pseudo-inverse. Many activation functions such as sigmoidal, sine, Gaussian or hard limiting functions can be used at the hidden layer and the class is determined as the class which has the maximum output value. A comprehensive description of the **S-ELM** algorithm (Huang et al., 2006) is given in the general introduction section of this thesis, under methods and optimization. Even though the **ELM** algorithm requires less training time, the random selection of input weights affects the generalization performance when the data is sparse or data is imbalanced. An improved version of **ELM** called the Sparse-ELM (**S-ELM**) (Suresh et al., 2010) which gives better generalization for sparse data, is used for predicting the **RSA** of proteins, where the imbalance in data varies with the different threshold values used. **S-ELM** is also well suited for **RSA** predictions of sequences whose structures have not yet been determined and for which there are no homologs in existing sequences.

The data used in this study is discussed in detail under the methods and data section. We call the **ELM** algorithm for each of the training data sets over several thresholds (degree of exposure to the solvent environment). We find the optimal number of hidden neurons using a unipolar sigmoidal activation function (lambda = 0.001) and perform K-fold ($K = 5$) validations. In K-fold validation, the training set is separated into K-groups. $K - 1$ groups are used for training in each of the K iterations and the model is tested on the remaining K^{th} group. The optimal parameters obtained during model building are stored and used during the testing phase. The performance of the **S-ELM** classifier and the time taken to develop the **S-ELM** model for **RSA** prediction are compared with the performance and time taken to process the same data using **SVM** algorithm. LIB**SVM** (Fan et al., 2005) software was used to determine the results for **SVM** approach. We show that the **S-ELM** algorithm can achieve better performance with much smaller processing times. Five-fold cross validation accuracies, processing time gains and comparative studies are further discussed in the results section.

Table 9.1 Number of residues per class for 2-class and 3-class data.

Thresholds for data were set between 0 and 50% for two class (C0 and C1) and between 10, 20, 25 and 50% for 3-class (C0,C1 and C2) data. The asterisk indicate null values where there is no class-3.

	No. of Training residues			**No. of Testing residues**		
%	**C**0	**C**1	**C**2	**C**0	**C**1	**C**2
0	867	6678	**	4713	38424	**
5	5796	1749	**	32943	10194	**
10	2826	4719	**	15864	27273	**
20	4065	3480	**	23111	20026	**
50	5796	1749	**	32945	10192	**
10:20	3888	831	2826	22265	5008	15864
25:50	1750	1750	4065	10194	9832	23111

9.1.2 Data generation for RSA prediction

Proteins consist of sequences of amino acid residues that play a key role in determining the secondary and tertiary structure of a protein. The sequential relationship among the solvent

accessibilities of neighboring residues can be used to improve the results (although solvent accessibility is considered evolutionarily less preserved than secondary structure). We use binary values and a window size of 8 to represent the amino acid sequences. **RSA** of an amino acid residue is defined (Mucchielli-Giorgi et al., 1999) as the ratio of the solvent-accessible surface area of the residue observed in the 3-D structure to that observed in an extended tri-peptide (Gly-X-Gly or Ala-X-Ala) conformation. **RSA** is a simple measure of the degree to which each residue in an amino acid sequence is exposed to its solvent environment. For our study, we consider the well-known Manesh data set (Naderi-Manesh et al., 2001) which has high imbalance with respect to the number of samples per class as given in Table 9.1, where the number of samples belonging to some classes is much less compared to the number of samples belonging to the other classes. The Manesh data set consists of 215 proteins, of which 30 proteins (7545 residues) with variable number of amino acid residues were used for classifier model development and the remaining 185 proteins (43137 residues) were used for evaluating the generalization performance of the **S-ELM** classifier through a 5-fold cross-validation model. The data in the training and testing set were cast into two-class and three-class problems 9.1 by determining whether the **RSA** value was below, between or above a particular threshold. We used various percentage-thresholds (0, 5, 10, 25, 50 for two-class and between 10_20 or 25_50 for three class), in order to compare our results with those existing in literature. A residue is considered as buried if its value is less than or equal to the lower range, partially buried if it is between the lower and the higher range and considered exposed if its **RSA** value is higher than the range of values (> 20 or > 50). The accuracy of the predictions depend on the value of the thresholds chosen and can vary widely with different residue compositions in different proteins as discussed in the results section.

9.2 Results and discussion

We compare the results of our simulation using **S-ELM** on the Manesh data set with the **SVM** algorithm and **NETASA** (Ahmad and Gromiha, 2002) methods Figure 9.1, using the same set of proteins for training and testing. Hence comparisons with literature are made only

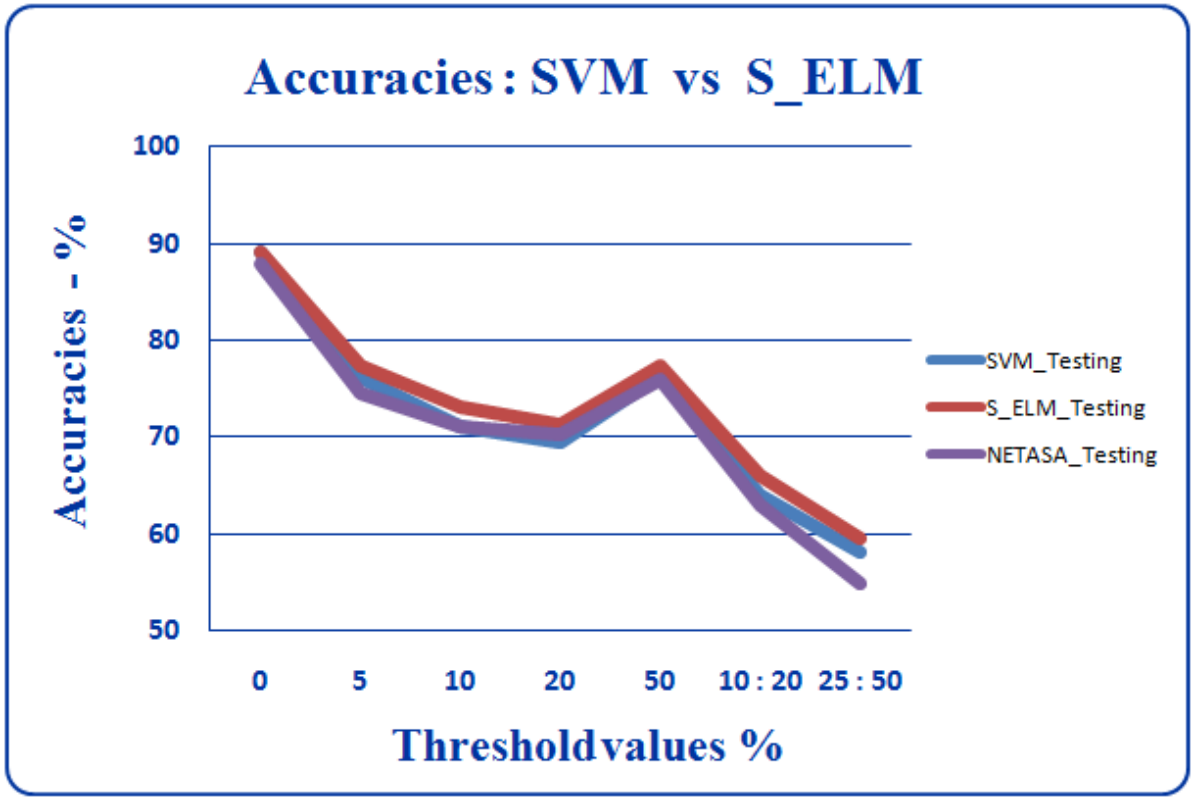


Figure 9.1 Accuracy comparison between **NETASA** , **SVM** and **S-ELM**

This figure shows improvements for **S-ELM** method. The results are discussed in detail in the results section.

with the **NETASA** results. The accuracy of the **RSA** predictions is measured by the number of residues correctly classified (positive class) as belonging to class1 for the two class problem and as belonging to class2 for the three class problem. Prediction accuracy for training and testing data sets is defined as the total number of correctly predicted values for each class over the total number of available residues. **S-ELM** approach achieves a better accuracy for training and testing than the corresponding results for the **NETASA** method for all sets of data as shown in Figure 9.1. The **SVM** algorithm takes a longer time to build the model as shown in Figure 9.2, whereas the **S-ELM** algorithm processes data at the same speed for all combinations of data, showing that the algorithm does not slow down when complex data is involved. **S-ELM** uses optimal parameters that are stored during the training phase making it possible to run through the tests quickly.

The training results for the **SVM** algorithm using this data set are at 99% for a range of thresholds. The corresponding test results for **SVM** vary from 69% to 89% over a range of thresholds for the two class problem. We see gains for all of the thresholds except for threshold of 20 where the accuracy is 69.5 which is slightly lesser when compared to **NETASA** results. The results are much better for the **S-ELM** algorithm, where the training and testing results are closer to each other showing better generalization. The training results vary between 73% and 89% for the 2-class problem, while the test results vary between 71% and 89% which are better than the results for the **SVM** and **NETASA** method. Our interest in including the **SVM** in our simulations was to show the advantages in time factor when the **S-ELM** algorithm is used. The training results for the **S-ELM** show a little gain over the **NETASA** and the **SVM** results, but the testing results for **S-ELM** clearly show higher results of between $.006$ to 4.476% as seen in Figure 9.1. Similarly for the three-class problem, seen on the last two lines of Table 9.2, the training accuracies for **SVM** are very high at 99% while the testing accuracies are 64% and 58% for two different thresholds, which are slightly higher than for the **NETASA** results.

For the **S-ELM** results, the training accuracies are closer to the testing accuracies, indicating better generalization for the 3-class problem also. Here the **S-ELM** test results show between 3 to 4% gains as compared to the **NETASA** results. As indicated by many results in the literature, the accuracies can vary widely for different thresholds and different number of classes into which the data is divided. A general trend in the literature is that the **RSA** prediction results vary between 70% and 80%, similar to what is seen here. So, the **S-ELM** gives comparable results to literature. These results can be further improved with optimization methods which can tune the parameters for the **S-ELM** and this will be a subject for our future studies. The main aim of this study is to show the efficiency with which **S-ELM** is able to process large amounts of data.

The biggest advantage of using **S-ELM** comes from the speed with which the data can be processed by the algorithm, while providing better accuracies. It can be seen from 9.2 that **S-ELM** has a clear advantage when it comes to processing speed. The same number of samples

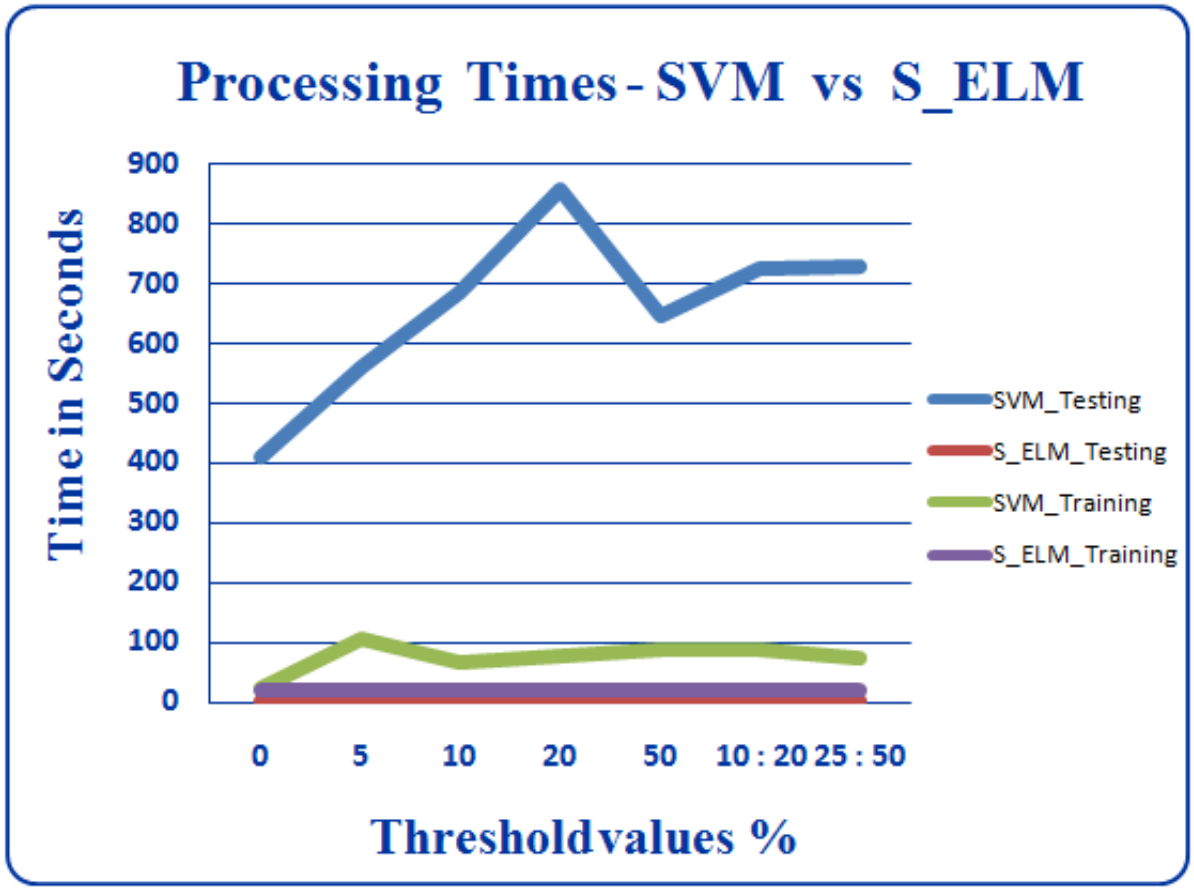


Figure 9.2 Processing time for training and testing: **SVM** Vs **S-ELM**.

This figure shows huge gains in processing time for **S-ELM** compared to **SVM**. These results are discussed in detail in the results section.

of 7545 training sample residues was used for model building for both algorithms. The ratio of time taken by **SVM** and **S-ELM** for model building, for the various thresholds range from 20.562 : 175 seconds which amounts to almost 8.51 times time gain by **S-ELM** for 0% threshold data. We find that the time gains range from 8 fold to multiple folds, the highest being for the 20% threshold data where the ratio is 20.562:1372.2 which is a 66.734 fold gain . Generally, the time taken for model building is most crucial, since the model needs to learn as much as possible in the shortest time. **S-ELM** will help to achieve these gains, which will be very important and necessary in view of the exponential increases we see in the availability of protein sequence information. For real time applications and for batch processing applications it

might be useful to have faster testing capabilities and here we see that the SELM algorithm is much faster in its testing capabilities also. The same number of 43137 testing residues was used for the test runs in both algorithms. Processing time ratio between the **SVM** and **S-ELM** algorithms for testing runs for 0% threshold is seen as .922:410, which amounts to 444.69 times faster processing by **S-ELM**. We find similar gains for other thresholds with the highest gain for the 20% threshold at .937:857 which is 914.62 times faster processing speed. Both the **SVM** and the **S-ELM** were run on the same computer running XP windows operating system with 4 GB RAM and MATLAB (Moler, 2011) software. Time taken for training and testing runs by **SVM** and **S-ELM** algorithms is given in Table 9.2 and Figure 9.2 illustrate the high processing time of **SVM** and the very low and steady processing times of **S-ELM** very clearly. The time taken by **S-ELM** is very low at less than one or two seconds, shown as a horizontal line close to the x-axis while the time taken by **SVM** is quite high, ranging between 200 and 1400 seconds for training and between 400 and 900 seconds for testing. **S-ELM** takes very little time for testing compared to the **NETASA** results, since stored optimal parameters are used to calculate the output analytically using ELM. As indicated by many results in the literature, the accuracies can vary widely for different thresholds and different number of classes into which the data is divided. A general trend in the literature is that the **RSA** prediction results vary between 70% and 80%, similar to what is seen here. So, the **S-ELM** gives comparable results to literature. There is no data known to us on processing times to compare speeds with the **NETASA** method. Future studies will concentrate on increasing the accuracy of **S-ELM** further using optimization techniques to tune the **S-ELM** parameters for **RSA** prediction.

9.3 Conclusions

We have used the **SVM** and **S-ELM** methods of classification for **RSA** prediction, using the Manesh data set. We have compared the performance of these algorithms with each other and with **NETASA** results, with respect to the speed of processing and have shown that there are multiple-fold gains in computational efficiency while using **S-ELM** algorithm. It will be advantageous to use the **S-ELM** algorithm for real time and batch processing applications

Table 9.2 Comparison between **SVM** and **S-ELM** processing times

	SVM			**S-ELM**		
	Time in Seconds			Time in Seconds		
Threshold%	**Modeling**	**Training**	**Testing**	**Modeling**	**Training**	**Testing**
0	175	24.6	410	20.6	0.5	0.92
5	990	105	561	20.9	0.6	0.94
10	1273	67	686	20.9	0.6	0.92
20	1372	76	857	20.9	0.5	0.94
50	977	89	645	20.9	0.6	0.95
10:20	1239	88	723	21	1.1	1.08
25:50	226	74	728	21	0.7	1.08

where accuracy and speed are equally important.

Acknowledgements

We acknowledge the support of National Institutes of Health through grants R01GM081680, R01GM072014, and R01GM073095 and the support of the NSF grant through IGERT-0504304.

REFERENCES

Adamczak, R., Porollo, A., and Meller, J. (2004). Accurate prediction of solvent accessibility using neural networks-based regression. *Proteins*, 56:753–67.

Ahmad, S. and Gromiha, M. (2002). NETASA: neural network based prediction of solvent accessibility. *Bioinformatics*, 18:819–24.

Berezin, C., Glaser, F., Rosenberg, J., Paz, I., Pupko, T., Fariselli, P., Casadio, R., and Ben-Tal, N. (2004). ConSeq: the identification of functionally and structurally important residues in protein sequences. *Bioinformatics*, 20:1322–1324.

Bondugula, R. and Xu, D. (2008). Combining sequence and structural profiles for protein solvent accessibility prediction. *Comput Syst Bioinformatics Conf*, 7:195–202.

Carugo, O. (2000). Predicting residue solvent accessibility from protein sequence by considering the sequence environment. *Protein Engineering Design and Selection*, 13:607–609.

Chen, H., Zhou, H.-X., Hu, X., and Yoo, I. (2004). Classification comparison of prediction of solvent accessibility from protein sequences. In *Proceedings of the second conference on Asia-Pacific bioinformatics - Volume 29*, APBC '04, pages 333–338, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

Cheng, J. and Baldi, P. (2006). A machine learning information retrieval approach to protein fold recognition. *Bioinformatics*, 22:1456–1463.

Cheng, J., Sweredoski, M., and Baldi, P. (2006). DOMpro : Protein Domain Prediction Using Profiles, secondary structure, relative solvent accessibility, and recursive neural networks. *Data Mining and Knowledge Discovery*, 13:1–10.

David, M. P. C., Asprer, J. J. T., Ibana, J. S. A., Concepcion, G. P., and Padlan, E. A. (2007). A study of the structural correlates of affinity maturation: Antibody affinity as a function of chemical interactions, structural plasticity and stability. *Molecular Immunology*, 44:1342–1351.

Fan, R., Chen, P., and Lin, C. (2005). Working set selection using second order information for training SVM. *Journal of Machine Learning Research*, 6:1889–1918.

Gianese, G., Bossa, F., and Pascarella, S. (2003). Improvement in prediction of solvent accessibility by probability profiles. *Protein Engineering*, 16:987–92.

Huang, G. B., Zhu, Q. Y., and K, S. C. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, 70:489–501.

Kim, H. and Park, H. (2004). Prediction of protein relative solvent accessibility with support vector machines and long-range interaction 3d local descriptor. *Proteins*, 54:557–62.

Meshkin, A. and Ghafuri, H. (2010). Prediction of Relative Solvent Accessibility by Support Vector Regression and best-first method. *Experimental and Clinical Sciences*, 9:29–38.

Moler, C. (1984-2011). *MATLAB:The Language of Technical Computing*. Mathworks, r2008a edition. http://www.mathworks.com/.

Mucchielli-Giorgi, M., Hazout, S., and Tuffery, P. (1999). PredAcc: prediction of solvent accessibility. *Bioinformatics*, 15:176–177.

Naderi-Manesh, H., Sadeghi, M., Araf, S., and Movahedi, A. (2001). Prediction of protein surface accessibility with information theory. *Proteins*, 42:452–9.

Nguyen, M. and Rajapakse, J. (2005). Prediction of protein Relative Solvent Accessibility with a two-stage SVM approach. *Proteins*, 59:30–7.

Ooi, T., Oobatake, M., Namethy, G., and Scheraga, H. (1987). Accessible surface areas as a measure of the thermodynamic parameters of hydration of peptides. *Proceedings of the National Academy of Sciences of the United States of America*, 84:3086–3090.

Petersen, B., Petersen, T. N., Andersen, P., Nielsen, M., and Lundegaard, C. (2009). A generic method for assignment of reliability scores applied to solvent accessibility predictions. *BMC Structural Biology*, 9:51.

Pollastri, G., Fariselli, P., Casadio, R., and Baldi, P. (2002). Prediction of coordination number and relative solvent accessibility in proteins. *Proteins*, 47:142–235.

Pollastri, G., Martin, A., Mooney, C., and Vullo, A. (2007). Accurate prediction of protein secondary structure and solvent accessibility by consensus combiners of sequence and structure information. *BMC Bioinformatics*, 8:201.

Shen, B. and Vihinen, M. (2003). RankViaContact: ranking and visualization of amino acid contacts. *Bioinformatics*, 19:2161–2162.

Sim, J., Kim, S., and Lee, J. (2005). Prediction of protein solvent accessibility using fuzzy k-nearest neighbor method. *Bioinformatics*, 21:2844–9.

Singh, Y. H., Gromiha, M. M., Sarai, A., and Ahmad, S. (2006). Atom-wise statistics and prediction of solvent accessibility in proteins. *Biophysical Chemistry*, 124:145–154.

Suresh, S., Saraswathi, S., and Sundararajan, N. (2010). Performance enhancement of extreme learning machine for multi-category sparse cancer classification. *Engineering Applications of Artificial Intelligence*, 23:1149–1157.

Wagner, M., Adamczak, R., Porollo, A., and Meller, J. (2005). Linear regression models for solvent accessibility prediction in proteins. *Journal of Computational Biology*, 12:355–69.

Zarei, R., Arab, S., and Sadeghi, M. (2007). A method for protein accessibility prediction based on residue types and conformational states. *Computational Biology and Chemistry*, 31:384–388.

PART IV

FLOPRED - FOR PHOSPHORYLATION PREDICTION IN PROTEINS

CHAPTER 10. FLOPRED METHODOLOGY FOR PREDICTION OF PHOSPHORYLATION SITES IN PROTEINS

Abstract

Phosphorylation is a post-translational modification on proteins to control and regulate their activities. It is an important mechanism for regulation of the biological functions in the body. Phosphorylated sites are known to be present often in intrinsically disordered regions of proteins that lack unique tertiary structures, and thus less information is available about the structures of phosphorylated sites. An important challenge is the prediction of phosphorylation sites in protein sequences obtained from mass-scale sequencing of genomes. Phosphorylation sites may aid in the determination of the functions of a protein or even differentiating mechanisms of protein functions in healthy and diseased states. **FLOPRED** is used to model and predict experimentally determined phosphorylation sites in protein sequences. Our new **PSO** optimization methods have enabled **FLOPRED** to predict phosphorylation sites with higher accuracy and with better generalization. Our preliminary studies on 984 sequences show that this model can predict phosphorylation sites with a training accuracy of 92.53% , a testing accuracy 91.42% and Mathews correlation coefficient of 83.9%.

10.1 Introduction

Phosphorylation often controls protein functions either by causing a change in structure or by changing charge of a binding site. This process activates and controls the reactions in a cell. Since phosphorylation sites are known to be in disordered regions, it is not always possible to detect these sites experimentally. It will be useful to have computational methods

to efficiently detect these sites.

10.2 Methods and data generation

FLOPRED methodology is used for predicting phosphorylation sites, where a neural network based Extreme Learning Machine is used for classification. The parameters of **ELM** are tuned by advanced Particle Swarm Optimization. The details of these methods can be seen in Section 1.4.1 and Section 1.4.2.

Data generation for phosphorylation prediction

$13,604$ sequences were obtained from the Phosphoto. **ELM** database (Dinkel et al., 2011), where experimental phosphorylation data found in literature has been stored for public use. In these sequences, a single residue is marked as a phosphorylated residue while all others are non-phosphorylated. If there are multiple phosphorylation sites, then they are given as two separate sequences. The phosphorylated sites are usually one of three residues, namely, serine, threonine or tyrosine, but for our preliminary study we are considering only two classes where a residue is either phosphorylated or not, i.e., we do not consider the type of residue that is phosphorylated.

The sequences in the data are coded using an orthogonal representation (binary coding) where each amino acid is represented as a twenty digit binary code, with the letter of interest being a 1 and the remaining letters denoted as a zero. A sliding window of 9 residues are used to represent the data. This data is then used for determining whether a residue is phosphorylated or not. Since only one residue in each sequence is marked as phosphorylated, there were 13604 residues with positive class for phosphorylation but many more residues which were in the negative class for unphosphorylated residues. To maintain a balance, the same number of residues were selected from each group for the classification using **FLOPRED** .

10.3 Results and discussion

1968 residues are used for each classification of which 984 residues are phosphorylated and 984 are not phosphorylated. Similarly, the test set also has 985 residues each in both classes. Our new **PSO** optimization methods have enabled **FLOPRED** to predict phosphorylation sites with higher accuracy and with better generalization. Our preliminary studies on 984 sequences show that this model can predict phosphorylation sites with a training accuracy of 92.53%, a testing accuracy 91.42% and Mathews correlation coefficient of 83.9%. In the testing results, the sensitivity for class-1 (non-phosphorylated) is 85.60% and specificity is 99.51% and Mathews corr-coeff is 83.97%, with accuracy of 99.59%. For class-2 (phosphorylated) residues, sensitivity is 85.60% and specificity is 85.60% and Mathews corr-coeff is 83.97% and accuracy was lower at 83.25%. So, the results for the negative class are higher than the results for the positive class (phosphorylated), although there are the same numbers of each class in our trials.

This experiment can be carried out in larger data sets with larger numbers of residues or as a multi-class problem where we can try to differentiate between various types of phosphorylation sites. The main aim of this study has been to show the robustness of the **FLOPRED** algorithm in being able to do good classification on sparse data and the results are promising.

10.4 Conclusions

FLOPRED methodology was used to classify phosphorylated and non-phosphorylated data with fairly high accuracy. Future studies may include larger data sets and multiple classification of different phosphorylation sites. These classifications can also be combined with the study of disordered regions that are more likely to contain phosphorylation sites.

REFERENCES

Dinkel, H., Chica, C., Via, A., Gould, C. M., Jensen, L. J., Gibson, T. J., and Diella, F. (2011). Phospho.ELM: a database of phosphorylation sites-update 2011. *Nucleic Acids Research*.

PART V

GENERAL CONCLUSIONS

CHAPTER 11. GENERAL CONCLUSIONS

11.1 Secondary Structure Prediction

Several secondary structure prediction methods are used in this study as summarized in as summarized in Section 8.1 on page 152 . While initial studies using Extreme Learning Machine and Particle Swarm Optimization yield good results, later studies using advanced and improved Particle Swarm Optimization techniques yield better results. Several other algorithms such as Genetic Algorithm and Principal Component Analysis are used for secondary structure prediction on a variety of data and these studies show promising results.

11.2 Relative Solvent Accessibility prediction

Support Vector Machines and Extreme Learning Machines are used for Relative Solvent Accessibility predictions on a set of protein sequences. The results of this study show results comparable to those in the literature. This study illustrates the increased speed with which Extreme Learning Machine is able to classify data. With the large number of available protein sequences, it is important to build algorithms that will be able to process data at faster speeds without compromising accuracy.

11.3 Prediction of phosphorylation sites

Prediction of phosphorylation sites using **FLOPRED** algorithm yields higher accuracies compared to those found in similar studies. This study illustrates that our algorithm is equally proficient at classifying binary data as it is in classifying real-number coded data that encodes better patterns to enable machine learning algorithms to differentiate between different

classes.

In summary, several algorithms such as **ELM, PSO, GA, PCA** and **SVM** were used in our studies. Several of their parameters were learned through the **PSO** approach. The results are significantly better than those found in literature for secondary structure prediction and phosphorylation prediction and comparable in quality for predicting the relative solvent accessibility.

Our contribution to secondary structure prediction is the **FLOPRED** algorithm which is simpler and faster. We have shown that **FLOPRED** can be used to predict secondary structures with better accuracies in a shorter period of time. These two aspects become increasingly important as huge numbers of protein sequences are available from the many genome sequencing projects and as increased numbers of sequences of individual organisms and individual humans are available for diagnostics of diseases.

APPENDIX A. List of template proteins used to generate profiles

Table A.1 List of 200 template proteins

. The given list of 422 proteins, in Table A.1 and Table A.2, were used to generate profile features for all the proteins used in this study. The list had to be separated into two tables in order to fit the page. The first 200 proteins are listed below. Table A.2 gives the remaining 222 proteins.

List of template proteins										
1	1a6m	1d4o	1g4i	1i4u	1k6u	1lug	1oa0	1pwg	1sen	1uai
2	1aho	1d4t	1g66	1i76	1k7c	1lwb	1oai	1pz4	1sfs	1ucr
3	1arb	1dbf	1ga6	1ifc	1ka1	1lzl	1od3	1q6o	1sg4	1ucs
4	1b9o	1e4m	1gqv	1ijv	1kf3	1m1q	1odm	1qft	1six	1ufy
5	1bkr	1e5k	1gve	1iqz	1kg2	1m4l	1oh4	1ql0	1sjw	1ug6
6	1bqk	1e9g	1gwe	1iro	1kmv	1m6z	1ok0	1qtw	1su8	1unq
7	1brf	1eaj	1gwm	1itx	1kng	1mj5	1olr	1r0r	1sxv	1uow
8	1bx7	1eb6	1gxm	1iua	1knm	1mn8	1ooh	1r2q	1t2d	1use
9	1byi	1et1	1gxu	1j0o	1kqp	1n0q	1oot	1r6j	1t3y	1uwc
10	1c5e	1euw	1gyx	1j8q	1kqw	1n1p	1oqv	1rb9	1t8k	1uz3
11	1c75	1exr	1h12	1jfb	1kt7	1n40	1p1x	1rg8	1tg0	1uzv
12	1c7k	1f1g	1h1n	1jg1	1kth	1n8v	1p4c	1rqw	1thm	1v0l
13	1c9o	1f41	1h4g	1jkv	1kug	1nki	1p5f	1rro	1tjy	1v6p
14	1cc8	1f94	1h97	1jm1	1kyf	1npi	1p6o	1rtq	1tkj	1vbw
15	1cex	1f9y	1hdo	1jo0	1l9l	1nww	1p9g	1rwy	1tqg	1vf8
16	1cse	1fcy	1hx0	1jo8	1lk2	1nwz	1pjx	1s0r	1tu9	1vh5
17	1ctj	1fd3	1hxh	1jr0	1lkk	1nxm	1plc	1s1p	1tuk	1vim
18	1ctq	1fk5	1hyo	1k3y	1lni	1nyk	1po7	1s5n	1u07	1vkk
19	1cy5	1flm	1i1x	1k4i	1lq9	1o7j	1pq7	1sau	1u1w	1vyr
20	1czp	1g2r	1i40	1k5c	1ls9	1o7q	1psr	1sby	1u2h	1w0n

Table A.2 List of 222 template proteins

. The given list of proteins, in A.1 and Table A.2, were used to generate profile features for all the proteins used in this study. The list had to be separated into two tables in order to fit the page. The first 200 proteins are listed in Table A.1. The remaining 222 proteins are given below.

List of template proteins - continuation of Table A.1										
1	1w23	1xmk	2akf	2ccv	2erl	2gxq	2ixt	2o0b	2qdy	2wea
2	1w2l	1xmt	2akz	2ccw	2f01	2hds	2izx	2o37	2qf4	2z4u
3	1w66	1xqo	2aqm	2chh	2f91	2heu	2j45	2o90	2qim	2z5w
4	1wbe	1xt5	2asc	2ciw	2fba	2hin	2j6b	2o9s	2qsk	2zex
5	1wc2	1y2k	2avm	2cl2	2fdn	2ho2	2j6l	2ofc	2r31	3b4u
6	1wcg	1yfq	2axw	2cov	2fe5	2hxm	2j8b	2okt	2r5o	3b5m
7	1wcw	1ys1	2b3h	2cs7	2fgo	2hxs	2j8w	2oln	2rb5	3b64
8	1wdd	1z2n	2b3n	2cws	2fma	2hyk	2j9c	2oss	2rdq	3b9w
9	1wdp	1z2u	2b82	2czq	2frg	2hys	2jcq	2ov0	2tps	3bc9
10	1wkr	1z53	2b97	2d5w	2fs6	2i24	2jda	2oxc	2uu8	3bfo
11	1wm3	1zk4	2bf6	2ddx	2ft6	2i49	2jek	2p02	2uuy	3bfq
12	1wma	1zl0	2bf9	2dfb	2fvv	2i4a	2jfr	2p5k	2v1q	3bmz
13	1wmd	1zlb	2bjd	2dkj	2fvy	2i5v	2jhf	2pgo	2v3g	3bqp
14	1wpn	1zuu	2bt9	2drm	2fwh	2i61	2lis	2phn	2v3i	3bs2
15	1wri	1zuy	2bv4	2dsx	2gb4	2i7d	2mhr	2pie	2v8t	3bxu
16	1wuk	1zzk	2bwf	2e3b	2gf3	2i8t	2nlr	2pmr	2v9v	3c2u
17	1wvf	2a26	2c2u	2e4t	2ggc	2ibl	2nn8	2pnd	2vb1	3c3y
18	1wyx	2a28	2c6z	2e5f	2gj3	2ic6	2nrl	2pne	2vba	3c70
19	1x1r	2a6z	2c71	2e6f	2gke	2igd	2nsz	2pwa	2vbk	3c8p
20	1x6i	2ab0	2c9v	2e7z	2gkg	2iim	2nuk	2q20	2vfr	3c8y
21	1x8q	2ahn	2cak	2ehz	2gqt	2imf	2nwd	2q3g	2vji	3ci3
22	1x9i	2aib	2car	2ekp	2gud	2imq	2nxv	2qcp	2vla	3cjs
										6fd1
										7a3h

APPENDIX B. List of target proteins used in the initial study

Table B.1 List of 40 target proteins

used in the initial study. The given list of 84 proteins, in Table B.1 and Table B.2, consisting of a total of 6635 residues, were used in the initial study. The list had to be separated into two tables in order to fit the page. The first 40 proteins are given below. The protein names and the number of residues in each protein are given.

List of proteins					
No.	Protein name	Num. of residues	No.	Protein name	Num. of residues
1	1aazb-1-DOMAK	87	21	1cc5	83
2	1acx	108	22	1cdlg-1-DOMAK	20
3	1adeb-2-AUTO.1	100	23	1cdta	60
4	1ahb-2-GJB	67	24	1cei-1-GJB	85
5	1amg-2-AS	57	25	1ceo-2-AUTO.1	53
6	1atpi-1-DOMAK	20	26	1cewi-1-DOMAK	108
7	1avhb-3-AS	86	27	1cfb-1-AS	101
8	1avhb-4-AS	74	28	1cgu-2-GJB	96
9	1ayab-1-GJB	101	29	1cgu-3-GJB	84
10	1bdo-1-AS	80	30	1cgu-4-GJB	104
11	1bds	43	31	1chbe-1-DOMAK	103
12	1bet-1-DOMAK	107	32	1chkb-2-AUTO.1	95
13	1bncb-1-AS	114	33	1cksc-1-AUTO.1	78
14	1bncb-3-AS	51	34	1clc-1-AS.1	102
15	1bncb-4-AS	118	35	1coi-1-AS	29
16	1bovb-1-DOMAK	69	36	1comc-1-DOMAK	119
17	1bpha-1-DOMAK	21	37	1crn	46
18	1brse-1-DOMAK	86	38	1csei	63
19	1bsdb-1-DOMAK	107	39	1ctf-1-DOMAK	68
20	1cbh	36	40	1cthb-1-DOMAK	79

Table B.2 List of 44 target proteins

used in the initial study. The given list of proteins, consisting of a total of 6635 residues, were used in the initial study. The list below is a continuation of Table B.1 and lists the remaining 44 proteins. The protein names and the number of residues in each protein are given.

List of target proteins (continuation of table B.1)					
No.	Protein name	Num. of residues	No.	Protein name	Num. of residues
41	1ctm-2-DOMAK	60	63	1gal-2-AS	116
42	1ctn-1-AS.1	109	64	1gcmc-1-AUTO.1	33
43	1ctn-3-AS.1	73	65	1gky-2-AS	50
44	1daab-1-AS	119	66	1gln-2-AS	116
45	1dar-3-AS	37	67	1gln-3-AS	48
46	1delb-2-AUTO.1	119	68	1gln-4-AS	98
47	1dfnb-1-DOMAK	30	69	1gmpb-1-DOMAK	96
48	1dih-2-AS	110	70	1gnd-2-JAC	97
49	1dsbb-2-AUTO.1	64	71	1gog-3-AS.1	98
50	1dynb-1-AUTO.1	113	72	1gp2a-1-AUTO.1	28
51	1ecl-4-AS	117	73	1grj-1-AS	74
52	1edmc-1-AUTO.1	39	74	1grj-2-AS	77
53	1edn-1-AS	21	75	1hcgb-1-AS	51
54	1eft-3-DOMAK	95	76	1hcra-1-DOMAK	52
55	1efud-2-AUTO.1	89	77	1hip	85
56	1euu-2-JAC	100	78	1hiws-1-AS	103
57	1fc2c	44	79	1hmy-2-AS	98
58	1fdx	53	80	1hnf-1	101
59	1fjmb-2-AS	111	81	1hnf-2	78
60	1fkf	107	82	1hplb2	111
61	1fuqb-3-AUTO.1	66	83	hslb2	102
62	1fxia	96	84	1htrp	43

APPENDIX C. List of target proteins used in the final study

Table C.1 A List of the first set of 120 proteins

. The list of 415 proteins shown in Tables C.1, C.2, C.3 and C.4, were used to generate profile features for all the proteins used in the final study. The list had to be separated into four tables in order to fit the page. The first set of 120 proteins are listed below. The remaining protein names are given in subsequent tables.

List of proteins - set 1 of 4					
1	1aazb-1-domak	1bncb-4-as	1clc-2-as.1	1dik-1-as.1	1fdt-1-as
2	1acx	1bovb-1-domak	1clc-3-as.1	1dik-2-as.1	1fdx
3	1ahb-2-gjb	1bpha-1-domak	1coi-1-as	1dik-4-as.1	1find-1-auto.1
4	1alkb-1-as	1bsdb-1-domak	1colb-1-domak	1din-1-as	1find-2-auto.1
5	1aorb-1-as	1cbg-1-as	1cpcl-1-domak	1dlc-1-as.1	1fkf
6	1aorb-3-as	1cbh	1cpn-1-domak	1dlc-3-as.1	1fnd
7	1aozb-1-as	1cc5	1cqa-1-auto.1	1dnpb-1-auto.1	1fua-1-auto.1
8	1aozb-2-as	1cdlg-1-domak	1crn	1dnpb-2-auto.1	1fuqb-1-auto.1
9	1aozb-3-as	1cdta	1csei	1dpgb-1-auto.1	1fuqb-3-auto.1
10	1atpi-1-domak	1cei-1-gjb	1cthb-1-domak	1dsbb-2-auto.1	1fxia
11	1avhb-3-as	1celb-1-auto.1	1ctm-2-domak	1dts-1-auto.1	1gal-2-as
12	1avhb-4-as	1cem-1-gjb	1ctn-1-as.1	1dupa-1-as	1gal-3-as
13	1ayab-1-gjb	1ceo-2-auto.1	1ctn-3-as.1	1dynb-1-auto.1	1gcb-2-as
14	1azu	1cewi-1-domak	1ctu-1-auto.1	1eca	1gcmc-1-auto.1
15	1bam-1-as	1cfb-1-as	1ctu-2-auto.1	1eceb-1-auto.1	1gd1o
16	1bbpa	1cfr-1-gjb	1cxsa-4-auto.1	1ecpf-1-auto.1	1gdj
17	1bcx-1-domak	1cgu-2-gjb	1cyx-1-auto.1	1edd-1-domak	1gep-3-as
18	1bdo-1-as	1cgu-3-gjb	1daab-1-as	1ese-1-auto.1	1ghsb-1-gjb
19	1bet-1-domak	1cgu-4-gjb	1daab-2-as	1etu	1gln-2-as
20	1bfg-1-domak	1chbe-1-domak	1dar-3-as	1euu-2-jac	1gln-3-as
21	1bmv1	1chd-1-as	1delb-2-auto.1	1fbab-1-domak	1gln-4-as
22	1bmv2	1chkb-2-auto.1	1dfji-1-auto.1	1fbl-1-as	1gmpb-1-domak
23	1bncb-1-as	1cksc-1-auto.1	1dfnb-1-domak	1fc2c	1gnd-2-jac
24	1bncb-3-as	1clc-1-as.1	1dih-2-as	1fdlh	1gog-1-as.1

Table C.2 A list of the second set of 120 target proteins

are given below. The remaining protein names are in subsequent tables.

List of proteins - set 2 of 4					
1	1gog-3-as.1	1hup-1-as.1	1lbu-1-as	1ndh-1-as	1pkyc-3-auto.1
2	1gp1a	1hvq-1-auto.1	1lbu-2-as	1ndh-2-as	1pnt-1-as
3	1gp2a-1-auto.1	1hxn-1-as	1lehb-3-as	1nfp-1-as	1poc-1-domak
4	1gp2g-2-as	1hyp-1-domak	1lib-1-domak	1nga-2-as.1	1powb-1-domak
5	1gpmd-4-as	1ignb-2-gjb	1lki-1-as	1nlkl-1-domak	1powb-2-domak
6	1gpmd-5-as	1il8a	1lmb3	1nox-1-gjb	1powb-3-domak
7	1grj-1-as	1ilk-1-as	1lpe-1-domak	1nozb-2-auto.1	1powb-4-domak
8	1grj-2-as	1ilk-2-as	1masb-1-auto.1	1oacb-2-as.1	1ppi-2-as
9	1gtmc-2-auto.1	1inp-2-as.1	1mcti-1-auto.1	1oacb-3-as.1	1ppt
10	1gtqb-1-auto.1	1irk-1-as	1mdaj-1-gjb	1oacb-4-as.1	1ptr-1-auto.1
11	1gym-1-auto.1	1irk-2-as	1mdam-1-domak	1onrb-1-auto.1	1ptx-1-as
12	1han-1-auto.1	1isab-1-gjb	1mdta-1-as	1otgc-1-as	1pyp
13	1han-2-auto.1	1isab-2-gjb	1mdta-2-as	1ovb-1-gjb	1pyta-1-as
14	1hcgb-1-as	1isub-1-domak	1mdta-3-as	1ovoa	1qbb-2-auto.1
15	1hcra-1-domak	1jud-1-gjb	1mjc-1-domak	1oyc-1-as	1qbb-3-auto.1
16	1hiws-1-as	1kinb-1-auto.1	1mla-2-as.1	1paz	1qbb-4-auto.1
17	1hjrd-1-auto.1	1knb-1-as	1mns-2-as	1pbp-2-domak	1qrdb-1-auto.1
18	1hmpb-1-auto.1	1kte-1-as	1mof-1-as	1pbwb-1-as	1r092
19	1hnf-1-as	1ktq-1-auto.1	1mrrb-1-domak	1pda-2-as	1rbp
20	1hnf-2-as	1kuh-1-as	1mspb-1-as	1pda-3-as	1rec-1-domak
21	1hplb-1-as	1l58	1nal4-1-auto.1	1pdnc-2-as	1rec-2-domak
22	1hplb-2-as	1lap	1nar-1-domak	1pdo-1-gjb	1regy-1-auto.1
23	1hslb-2-domak	1latb-1-auto.1	1nbac-1-as	1pht-1-auto.1	1reqc-1-as
24	1htrp-1-as	1lba-1-domak	1ncg-1-auto.2	1pii-2-domak	1reqc-2-as

Table C.3 A list of the third set of 120 target proteins

are given below. The remaining protein names are in the next table.

List of proteins - set 3 of 4					
1	1rhgc-1-domak	1svb-1-as	1vcab-1-auto.1	2bopa-1-domak	2ltnb
2	1rie-1-gjb	1svb-2-as	1vcab-2-auto.1	2cab	2mev4
3	1ris-1-domak	1tabi-1-domak	1vhh-1-as	2ccya	2mltb-1-gjb
4	1rlds-1-domak	1taq-2-as	1vhrb-2-auto.1	2cmd-2-gjb	2mtac-1-as
5	1rlr-1-jac	1tcba-1-as	1vid-1-jac	2dkb-2-as	2nadb-2-as.1
6	1rlr-2-jac	1tcra-2-gjb	1vmob-1-as	2dln-1-as	2npx-3-as.1
7	1rsy-1-as	1tfr-1-gjb	1vpt-1-jac	2dln-3-as	2or1l
8	1rvvz-1-auto.1	1thtb-1-auto.1	1wapv-1-auto.1	2dnja-1-as	2paba
9	1s01	1thx-1-auto.1	1wfbb-1-auto.1	2ebn-1-as	2pgd-1-auto.1
10	1scud-1-as	1tie-1-domak	1wsya	2fox	2pgd-2-auto.1
11	1scue-2-as	1tif-1-as	1wsyb	2gbp	2phh
12	1scue-3-as	1tiic-1-gjb	1yptb-1-auto.1	2gcr	2polb-1-as
13	1seib-1-auto.1	1tml-1-as	1yrna-2-as	2glsa	2reb-1-domak
14	1sesa-2-as	1tnfa	1znbb-1-as	2gn5	2rspa
15	1sfe-1-as	1tplb-3-as	2aat	2gsq-2-as	2scpb-1-domak
16	1sfe-2-as	1trb-2-as	2abk-2-as	2hft-1-as	2sns
17	1sftb-2-as	1trh-1-as	2admb-1-auto.1	2hft-2-as	2sodb
18	1sh1	1trkb-1-as	2admb-2-auto.1	2hhmb-1-domak	2spt-1-domak
19	1smpi-1-as	1trkb-3-as	2afnc-1-auto.1	2hhmb-2-domak	2spt-2-domak
20	1spbp-1-as	1tsp-1-as	2afnc-2-auto.1	2hipb-1-domak	2stv
21	1sra-1-as	1tssb-2-domak	2alp	2hmza	2tgi-1-domak
22	1srja-1-domak	1ubq	2asr-1-domak	2hpr-1-domak	2tmdb-3-as
23	1stfi-1-domak	1udh-1-auto.1	2bat-1-gjb	2i1b	2tmvp
24	1stme-1-auto.1	1umub-1-as	2bltb-2-auto.1	2ltna	2trt-1-auto.1

Table C.4 List of final set of 55 target proteins.

List of proteins - set 4 of 4			
1	3ait	3tima	7icd
2	3blm	4bp2	821p-1-domak
3	3cd4	4gr1	8adh
4	3chy-1-domak	4pfk	9apia
5	3cln	4rhv1	9apib
6	3cox-1-as.1	4rhv3	9insb
7	3cox-2-as.1	4rhv4	9pap
8	3ecab-1-as	4rxn	
9	3ecab-2-as	4sdha	
10	3gapa	4sgbi	
11	3hmga	4ts1a	
12	3hmgb	4xiaa	
13	3icb	5cytr	
14	3inkd-1-domak	5er2e	
15	3mddb-1-as	5ldh	
16	3mddb-2-as	5lyz	
17	3mddb-3-as	6cpa	
18	3pgk-2-as	6cpp	
19	3pgm	6cts	
20	3pmgb-1-as	6dfr	
21	3pmgb-2-as	6hir	
22	3pmgb-3-as	6rlxd-1-domak	
23	3pmgb-4-as	6tmne	
24	3rnt	7cata	

APPENDIX D. Definitions of secondary structure accuracy measures

All of the discussions below are under the assumption that α-helix is the positive class and β-sheet and coil are the negative classes, where the results are combined for these latter two classes. Similar arguments can be made by considering the other two classes as the positive class. The quantities defined below were used for calculating the post-test odds in Section 2.1.7.1 on page 36 .

Four quantities are used to calculate the accuracy measures. They are defined as follows:

1. *True-Positive (***TP***)*: If an amino acid residue belongs to one of three secondary structure classes, say α-helix and is classified as α-helix, then the result is a *True-Positive*.

2. *False-Positive* (**FP**): When a sample in the negative class is classified as an α-helix in the positive class, then the result is a *False-Positive*. The classification algorithm is said to have *poor Specificity* if there are many *False-Positive* classifications.

3. *False-Negative (***FN***)*: If a residue is classified as one of the other two negative classes when it is in fact an α-helix, then it is a *False-Negative*. The classification algorithm is said to have *poor Sensitivity* if there are many residues which are *False-Negative*

4. *True-Negative (***TN***)*: If a residue belongs to a negative class and is correctly classified as a negative class, then the result is said to be *True-Negative*.

These four values **TP, TN, FP** and **FN** are used to calculate several metrics such as Sensitivity, Specificity etc., in order to determine the quality and reliability of our classification results. These metrics in turn are used in calculating additional metrics discussed below. All these metrics are used to discuss the results in later chapters, to determine the quality of our

training models which clearly have an impact on the predictions. In our classification results, the *observed classes* are the classes to which residues actually belong while the *predicted classes* are assigned by the classification algorithm.

Table D.1 Sensitivity, Specificity and other metrics.

This table shows the confusion matrix summary values that are used to determine the reliability of the predictions. These metrics are used to discuss the results of all classifications in later chapters. All terms and abbreviations used in this table are formulated and discussed in Section 2.1.7.

		Observed		
		Positive	**Negative**	
Predicted	**Positive**	TP	FP (Type I error)	$\rightarrow$ PPV
	Negative	FN (Type II error)	TN	$\rightarrow$ NPV
		Sensitivity	**Specificity**	

Specificity

Specificity gives the proportion of samples that belong to the negative classes (β-sheet and coil) that are identified correctly. If the Specificity is high, then it means that the algorithm is better able to identify a sample as belonging to the negative class (β-sheet or coil) and vice versa. These are called **FP** errors and are said to be *Type I errors or α errors*. Specificity is calculated as:

$$\textbf{Specificity} = \frac{TN}{TN + FP} \tag{D.1}$$

False Positive Rate

False Positive Rate **(FPR)** = 1 - Specificity; **(FPR)** is used to determine other summary metrics. Here, a residue that does not belong to the positive class, is classified as positive; since this is part of the learning process, this decision does not lead to the building of a good model. We are dismissing a result that is important for classifying a residue as a negative

class, and also misleading the learning process by classifying it as a positive class. We are saying something is true when it is not. So, we need to minimize these errors.

$$\mathbf{FPR} = \frac{FP}{FP + TN} \tag{D.2}$$

Sensitivity

Sensitivity gives the proportion of samples that belong to the positive class (α-helix) that are identified accurately. When the algorithm has good generalization, it is able to positively identify the observed class of the residue of interest and is said to have high Sensitivity. If the algorithm has *poor Sensitivity*, samples belonging to the positive class are classified as belonging to one of the negative classes (β-sheet and coil). These are called **FN** errors and are said to be *Type II errors or β error* with poor Sensitivity.

$$\mathbf{Sensitivity} = \frac{TP}{TP + FN} \tag{D.3}$$

False Negative Rate

False Negative Rate **FNR** = 1 - Sensitivity. **FNR** is used to determine other summary metrics. Here, a residue that belongs to the positive class, is classified as a negative class; since this is part of the learning process, it will negatively affect the model. We are dismissing a result that is important for classifying correctly, and misleading the learning process, by indicating that something is false when it is in fact true. So, we need to minimize these errors. **FNR** can be defined as:

$$\mathbf{FNR} = \frac{FN}{TP + FN} \tag{D.4}$$

In order to have a good balance in our classification results, we want a good balance between Sensitivity and Specificity. They should not be too high or two low. An algorithm that is able to achieve this will provide good generalization performance. A drawback to using these measures is that they do not consider all four metrics $TP, TN, FP and FN$ *simultaneously*

in their calculations, which results in an unbalanced view of the quality of the results. Hence we need better measures of accuracy and they are discussed below.

There are several statistics that help us to estimate the reliability of our classifications. Sensitivity and Specificity are not strongly useful for this purpose. *After obtaining the results*, we should be able to determine reliability measures using predicted values. It is useful to look at our results with respect to each of these metrics in order to assure ourselves that we have built a good model and have used representative data in the training and testing data sets. This information could also help to determine whether existing models might benefit from adjustments or improvements when we want to classify new sequences.

Positive Predictive Value PPV

This statistic gives the proportion of secondary structures that are correctly classified as being in the positive class (α-helix). A high **PPV** means that only in rare cases will a positive class be classified as being a β-sheet or a coil and that these results are highly reliable. The metric **PPV** *does not say anything about how often the algorithm is misclassifying a negative class (β-sheet or a coil) in the positive class*, which is clearly a limitation. In these cases we cannot calculate the reliability of predictions for the negative classes. **PPV** is calculated as:

$$\mathbf{PPV} = \frac{TP}{TP + FP} \tag{D.5}$$

Negative Predictive Value NPV

This statistic gives the proportion of secondary structures that are correctly classified as being in the negative class (β-sheet and coil). A high **NPV** means that the results for negative classifications are highly reliable and only in rare cases, is a negative class being classified in the positive class (α-helix. The **NPV** *does not say anything about how often the algorithm misclassifies a positive class as being a negative class*,which is clearly a limitation. In these cases we cannot calculate the reliability of predictions for the positive classes. **NPV** is calculated as:

$$\mathbf{NPV} = \frac{TN}{TN + FN} \tag{D.6}$$

However, the **PPV** and **NPV** *critically depend on the prevalence* (% content) of the secondary structures in the training model. If the occurrences of positive cases are rare, then negative classifications are *expected to be* more reliable and more informative while the positive classifications are expected to be less reliable and vice versa. But in our analysis of the results with respect to amino acid residues, we find that some residues with lower occurrences are classified with higher accuracies and vice versa. There are some factors that need to be considered in order to build a good model. The prevalence of the three secondary structures or the amino acids in the training data sets, might be different from the proportions in which they occur naturally. This situation might lead to misleading estimates of the predictive capabilities of a given algorithm. In addition, the test samples must be representative of the samples used for training, in order for these statistics to be useful metrics of the reliability of the classifications. If it is not possible to meet these requirements, then post-test probabilities, which are discussed next, can give a better idea about the reliability of the classifications. The model must be updated when newer, hitherto unrepresented proteins are included in the test sets, in order to maintain generalization performance.

The predictive values discussed above are general measures of accuracy. The post-test probability is a measure for classification of individual samples. These probabilities might be useful to fine tune classification results such as correcting predictions at the ends of secondary structures, which are error prone and difficult to classify. If the pre-test probability (of occurring in a secondary structure) of a sample in the test set, is the same as for similar samples in the training model, then the post-test probability and pre-test probability will be the, otherwise they will differ. So, there is a dependence on the prevalence of the secondary structures in the training models and the reliability of the results. In order to obtain a reliable estimate of **PPV** and **NPV** for any model, we can include the prevalence of the secondary structures (in the training model) in our calculations. If this prevalence is different from the value with which they occur naturally, then this difference can also be taken into account

when calculating some of the metrics as discussed below.

Positive Predictive Value with Prevalence

Positive Predictive Value with Prevalence (**PPVPrev**) can be defined as:

$$\mathbf{PPVPrev} = \frac{(Sensitivity * Prevalence)}{(Sensitivity * Prevalence) + ((1 - Specificity) * (1 - Prevalence))} \quad \text{(D.7)}$$

Negative Predictive Value with Prevalence

Negative Predictive Value with Prevalence (**NPVPrev**) can be defined as:

$$\mathbf{NPVPrev} = \frac{(Specificity) * (1 - Prevalence)}{(Specificity * (1 - Prevalence)) + ((1 - Sensitivity) * Prevalence)} \quad \text{(D.8)}$$

When there are a large numbers of false positives or false negatives, the Sensitivity and Specificity values cannot be used reliably to determine the effectiveness of the algorithm. The Likelihood Positive Ratio and Likelihood Negative Ratio are better measures since they simultaneously consider all four basic metrics, **TP, TN, FP** and **FN**. The likelihood Ratio of a test can help to estimate the post-test probabilities with the help of Pretest Odds. It can give an estimate of how much the test result themselves can influence the odds of a sample belonging to a positive or negative class. These types of estimates might help to determine the quality of the predictions and may be used to fine tune the classification results or correct errors that occur at the ends of secondary structures.

Likelihood Ratio Positive (LRP)

When a sample is classified as a member of a positive class, **LRP** (LR+ve) can give an estimate of how much the odds of the sample being in the positive class has changed *after the test results are obtained*. **LRP** is defined as:

$$\mathbf{LRP} = \frac{Sensitivity}{1 - Specificity} \quad \text{(D.9)}$$

Likelihood Ratio Negative (LRN)

Likelihood Negative Ratio (**LRN**): When a sample is classified as a member of the negative class, **LRN** (LR-ve) can give an estimate of how much the odds of the sample being in the negative class has changed *after the test results are obtained*. **LRN** is defined as:

$$\mathbf{LRN} = \frac{1 - Sensitivity}{Specificity} \tag{D.10}$$

LRP and **LRN** can be combined with the prevalence of secondary structures, characteristics of the training model and other information about the particular secondary structure to determine the post-test odds of the particular structure belonging to a class. The reliability of the classification results can be determined using these metrics. Initially, the pre-test odds, which is the likelihood of a particular sample belonging to the positive class(prevalence), prior to testing, is determined. This value, which can also be expressed as a probability, can be adjusted depending on the nature of the training samples that are used for the building the model.

The pre-test odds can be combined with **LRP** and **LRN** to obtain the post-test odds.

Something about myself!

Saras Saraswathi received the B.A. degree in Mathematics from University of Delhi, India, (1977), M.S. in Computer Science from Old Dominion University, USA, (1985) and did research in Artificial intelligence under Prof. Chris Wild at ODU. She was a member of the Voluntary Research Program at NASA, Langley Research Center (1985) and was involved in Space Station animations. She was an IT consultant in Bangalore, India ($1985 - 1991$). She was an IT tutor at NanyangTechnological University and National University of Singapore ($1991 - 2000$). She was a visiting Researcher; at the GINTIC Institute at Nanyang Technological University ($1992 - 1993$); at the Center for Networking and Excellence, Amrita University, India, ($2004 - 2006$) and at the Bioinformatics Research Center, Nanyang Technological University, Singapore ($2006 - 2007$), doing research in Bioinformatics using Machine Learning techniques. She received M.S. in Bioinformatics from Nanyang Technological University, Singapore, in 2007.

She is currently doing a PhD candidate in Bioinformatics at the Jernigan Laboratory, BBMB, Iowa State University (Saraswathi et al., 2011; Suresh et al., 2010; Saraswathi et al., 2010a,b). She has been awarded the National Scinece Foundation IGERT Traineeship in Computational Molecular Biology. Her current research interests are in predicting protein secondary structures using physical properties of amino acids and Machine Learning techniques.

She has been a member of IEEE since 1997 and is currently a member of IEEE-Computer Society, IEEE-Computational Intelligence Society, IEEE-Women in Engineering and the International Society for Computational Biology. She has been actively involved with ISCB-Student Council activities since 2006 and has been instrumental in starting several Regional Student Groups (RSGs) for Computational Biology around the world, almost 20 functioning

now. RSG-India alone has over 1500 members. She has been a member of the ISCB Education Committee and has contributed as a program committee member for Computational Intelligence conferences and ISCB-SC symposiums since 2006.

I hope to continue my research in bioinformatics and will strive to contribute meaningfully to the world of medicine for the benefit and well being of my fellow humans.

http://ribosome.bb.iastate.edu/saraswathi_s.html

RECENT PUBLICATIONS

Saraswathi, S., Jernigan, R. L., and Kloczkowski, A. (2010a). An Extreme Learning Machine Classifier for prediction of relative solvent accessibility in proteins. *Proceedings of IJCCI/ICNC*, pages 364–369.

Saraswathi, S., Jernigan, R. L., Koliniski, A., and Kloczkowski, A. (2010b). Protein secondary structure prediction using knowledge-based potentials. *Proceedings of IJCCI/ICNC*, pages 370–375.

Saraswathi, S., Suresh, S., and Sundararajan, N. (2011). Icga-pso-elm approach for accurate multiclass cancer classification resulting in reduced gene sets in which genes encoding secreted proteins are highly represented. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 8:452–463.

Suresh, S., Saraswathi, S., and Sundararajan, N. (2010). Performance enhancement of extreme learning machine for multi-category sparse cancer classification. *Engineering Applications of Artificial Intelligence*, 23:1149–1157.

DEDICATION

I would like to dedicate this thesis to ....

Maatha, Phitha , Guru and Deivam...... *Hindu parambaryam (tradition)* dictates that this is the order in which you should worship those who are most dear to you.

Maatha: My **MOTHER, Mrs. Vasanthi Rajagopal**, Coimbatore, India, who brought me into this world. There is no talk of anything else if you yourself do not exist.

Phitha: My **FATHER, Mr. M.J. Rajagopal**, Coimbatore, India.

Guru: My teachers... the selfless persons who share what they have, without fear that you might become someone bigger than themselves; who, irrespective of their own successes, want to see their student's succeed; who are at the foot of the ladders, helping students achieve heights; the giants on whose shoulders students stand and succeed. In my world, my GURUS are those whom I wish to emulate and follow by example. I dedicate this thesis to my GURUS who believed in me and helped me reach this point in life. This would include my very first teacher in my nursery school to all the wonderful professors in Iowa State where I stand proudly today.

Deivam: is the deity or God whom you worship. Deity is listed last but not least. It is difficult to thank someone without saying what they have done for you. So, Deivam comes after mentioning the others who were sent by God to help you. To me this is Lord Srinivasa himself who lives on the Seven Hills in India and Lord Ganesh who stood by me and helped me overcome the many obstacles I had to overcome before getting to this point in life, when my dreams seem like they are going to become a reality soon.