Species-Specific Protein Secondary Structure Prediction

By

Mariana Barssoum

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

Masters of Applied Science

in Biomedical Engineering

Ottawa-Carleton Institute for Biomedical Engineering

Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario, Canada
October 2009

Your file Votre référence
ISBN: 978-0-494-63819-4
Our file Notre référence
ISBN: 978-0-494-63819-4

Abstract

Protein secondary structure prediction methods aim to accurately predict the structure of a protein given knowledge only of its primary sequence. In this thesis, we investigate a new approach to the prediction of the protein secondary structure which creates species-specific predictors instead of using a single structure predictor trained using data pooled from multiple species. The underlying hypothesis that protein folding is influenced by species-specific differences is first investigated through a comparison of protein chain sequence and structure composition for 12 species representing all six Kingdoms of life. Next, various neural networks are trained with species-specific data to determine if there exists a particular neural network architecture that yields optimum prediction accuracy for a particular species. Through evaluation of five different network architectures, results show that the performance of Elman networks surpass other network architectures for most of the species. Elman networks are then trained with species-specific sequence and structure data. Five-fold cross-validation results over 12 species reveal that species-specific predictors are more effective than predictors trained on protein data pooled from multiple species. Interestingly, when an exact match between the test and train species is not available, results over 16 new species indicate that there is preference for predictors trained on phylogenetically related species. Lastly, we show that voting among several species-specific classifiers provides the highest classification accuracy. To my knowledge, this work represents the first investigation of species-specific neural network protein secondary structure prediction systems.

Acknowledgements

I would like to thank my supervisor, Professor James R. Green, for his great support and great efforts throughout my thesis. I benefited from his valuable advices and directions in almost all the work done in this thesis. I also thank him for his careful editing to this thesis. I thank him for his great patience and support not only in this thesis, but also for other courses that I studied in the past two years and also guiding me in learning Perl, Matlab, and C++.

Also, I would like to thank Professor Monique Frize, Professor WonSook Lee, and Professor Adrian Chan for their great efforts in the courses that I attended in the last two years.

Special thanks to my parents, my mother and the soul of my father, who supported me and helped me throughout my life. I will always remember that they have deprived themselves from all luxuries in life to enable me to study in one of the best private English schools. I thank them for all their precious love and kindness. I also thank my brother for being so supportive, so enthusiastic, and so positive about my thesis.

I would like to thank my beloved daughter and son, little angels Miriam and Anthony, which their presence in my life give me all the joy and happiness, and their smiles were enough to let me forget all the hard work and encourage me to continue studying.

I express all the gratitude to my husband, Medhat, who was very patient, very supportive, and very encouraging. I really appreciate all his help, and I cannot deny that without his continuous support, I would never have completed this thesis.

I really thank God for his great love and kindness, for giving me this great chance to study my Masters, for allowing me to do my Masters with a very kind and helpful Professor, and for giving me the best family ever.

This study was supported by the Natural Sciences and Engineering Research Council of Canada.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

1D	One-dimensional
3D	Three-dimensional
AA	Amino acid
ANN	Artificial neural network
BLAST	Basic local alignment search tool
DNA	Deoxyribonucleic acid
BRNN	Bidirectional recurrent neural network
BSMRNN	Bidirectional segmented memory recurrent neural network
DSSP	Dictionary of secondary structure assignments of proteins
HMM	Hidden Markov model
KNN	K-nearest neighbour
MSE	Mean squared error
N-CV	N-fold cross-validation test protocol
NMR	Nuclear magnetic resonance (spectroscopy)
NN	Nearest-neighbour
mRNA	Messenger ribonucleic acid
PDB	Protein data bank
PSI-BLAST	Position specific iterated BLAST
PSSM	Position specific scoring matrix
Q_3 score	Percent of residues correctly predicted in one of three states

1 Introduction

Proteins are the fundamental building units of living organisms and the work horses of most biochemical processes that permit life. Understanding the three-dimensional structure of a protein is crucial to understand its function and has extremely great importance in drug design, protein engineering, and also is fundamental in developing novel diagnostic methodologies. For this reason, it has been of keen interest to determine protein structure and function. The success of genome sequencing projects has resulted in an exponential increase of sequence data. However, structure data has been much slower to accumulate. As a result, much work has been done by researchers in order to reduce the sequence-structure gap. Although experimental determination of three-dimensional structure of proteins may provide high resolution structural models, these experimental approaches are very complex, time consuming and can only be applied to a subset of all proteins [1]. Consequently, computational three-dimensional structure prediction has attracted many researchers [1]. However the direct prediction of protein tertiary structure is a very challenging problem, thus many researchers have focused on the intermediate step of secondary structure prediction whose results can then be employed as a starting point to predict the tertiary structure. Many approaches to predict protein secondary structure have been explored including support vector machines[2], methods based on information theory techniques (for example GOR method [3][4]), neural networks [5][6][7][8], and nearest neighbour methods [9]. In this thesis, only neural networks will be discussed as they very popular methods and have achieved very high prediction accuracies [10][11].

Despite all the attempts to predict protein secondary structure, there is no reported work done to assess the effect of training neural networks with species-specific sequence and structure data rather than training them using data pooled from mixed species. Thus, in this

thesis the effect of training neural networks with species-specific datasets will be investigated and compared to neural networks trained with pooled data from mixed species. Also, different neural network architectures will be employed to evaluate if specific species would yield better accuracies using certain neural network architectures. Finally, classifiers trained on species-specific datasets will be evaluated to examine the optimum classifier for each species, and voting among several species-specific classifiers will be shown to have the highest accuracy.

1.1 Hypothesis and Goals

We seek to develop and evaluate species-specific protein secondary structure prediction systems. In this thesis, we hypothesise that species-specific neural network predictors will be more effective than predictors trained on protein data pooled from multiple species. The underlying assumption that there are species-specific aspects of protein folding that affect the mapping from primary sequence to secondary structure will first be investigated through a comparison of protein sequence and structure composition across multiple species. Also we hypothesise that when a perfect match between training and testing species is not available, classifiers trained using data from phylogenetically similar species will perform the best. Phylogenetics studies evolutionary relationships between species based on genetic differences accumulating over time. Phylogenetic distance measures the degree to which two species are genetically similar.

1.2 Thesis organization

This thesis consists of 7 chapters. Chapter 2 briefly discusses protein structure, its hierarchical levels, and describes how three dimensional protein structure can be determined through complex experimental procedures, such as X-ray crystallography and nuclear magnetic resonance, and through computational three-dimensional structure prediction methods. The importance of predicting protein secondary structure will be

discussed. Lastly, the historical development of neural networks will be discussed, from the early attempts to the state of the art of using neural networks to predict protein secondary structure.

In chapter 3, we will test hypothesis that there exists species-specific aspects to protein folding. This hypothesis will be investigated through a comparison of protein sequence and structure composition, and variation of protein chain lengths across multiple species.

Chapter 4 investigates the optimal network architecture for different species. Five different neural network architectures will be trained on 12 different species representing the six Kingdoms of life to investigate whether a particular neural network architecture would yield optimum prediction accuracy for a particular species. A brief description of the architecture and learning algorithms of each neural network is also presented.

In chapter 5, we will examine hypothesis that protein secondary structure prediction methods trained on species-specific datasets will outperform classifiers trained on mixed species datasets. Furthermore, we will explore the hypothesis that structure prediction accuracy will be highest when there is a close phylogenetic relationship between training and testing species. Lastly, we will demonstrate that voting among several species-specific classifiers provides the highest structure prediction accuracy.

Chapter 6 presents a summary of contributions and provides recommendations for future work.

2 Literature Review

This chapter will introduce the hierarchical levels of protein structure. Then, a quick review of experimental and prediction methods for the determination of protein structure will be discussed. After that, the importance of protein secondary structure prediction will be reviewed. Then different neural network architectures and learning algorithms will be discussed, with emphasis on their application to the prediction of protein secondary structure.

2.1 Protein structure

2.1.1 The Flow of Genetic Information

The fundamental building units and work horses of life are proteins [12]. According to the Human Genome Project, the human body contains approximately 100,000 different proteins [13][14]. Proteins are the building units of cells where they participate in almost all biochemical processes in cells. For example, the enzymes that are responsible for almost all the chemical transformations that occur in cells are, in fact, proteins. For this reason, it is of keen interest to many scientists to study the structure and function of proteins. Protein synthesis is performed in two steps that follow the central dogma of molecular biology as shown in Figure 2-1. The first step is the transcription process in which the deoxy-ribonucleic acid (DNA) is transcribed into messenger ribonucleic acid (mRNA). The second step involves the translation process in which proteins are synthesised by ribosomes from the 'recipe' encoded in the mRNA. The flow of genetic information in a cell is in one way from DNA to RNA to protein [15].

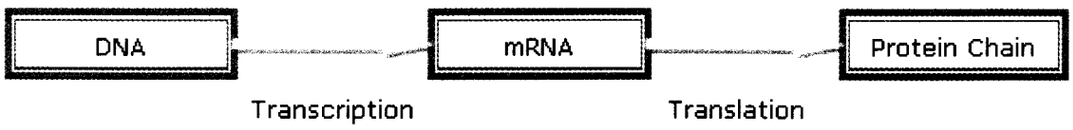

Figure 2-1: Protein synthesis.

DNA is a linear polymer composed of four monomers. It has a fixed back bone that is made up of a repeated sugar (deoxyribose) and phosphate units and one of four bases: adenine (A), cytosine (C), guanine (G), and thymine (T). The specific order of bases in a gene along a strand of DNA encodes the genetic information that carries instructions for assembling proteins [16]. DNA normally exists in a double helix formed by the combination of two DNA strands, and hydrogen bonds are formed between adenine from one DNA strand with thymine from the other strand, or between guanine from one DNA strand with cytosine from the other strand. Genes are regions in the DNA that encode proteins. A simple representation of the two strands of DNA double helix are shown in Figure 2-2.

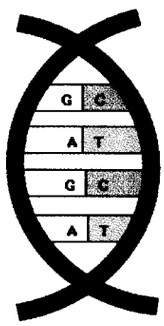

Figure 2-2: The two strands of DNA double helix.

The genome is composed of 46 chromosomes in every cell, where a chromosome is an organelle that is responsible for duplication and evolution of DNA. Proteins are linear polymers composed of 20 amino acids. Each amino acid is encoded by three bases along a DNA strand according to the genetic code. The sequence of amino acids determines the native three-dimension protein structure and the native three-dimension protein structure largely dictates the protein function [17].

2.1.2 The hierarchical levels of protein structure

Proteins are long polymers composed of a repetitive back bone of constant structure and side chains of variable composition that are attached to each residue. Proteins fold into a wide variety of patterns at different levels of granularity. This hierarch of patterns was defined by Linderstrom-Lang and Schellman [18] as primary structure, secondary structure, and tertiary structure. The term *quaternary structure* was introduced by *Bernal* in 1958 to refer to proteins that are composed of more than one subunit [19][20]. Each level of the hierarchy is defined in the following sections.

2.1.2.1 Primary Structure

The primary structure is unique for each protein. It refers to the order in which the amino acids are attached to one another. Proteins are composed of 20 amino acids. A list of names of the 20 AAs abbreviated with a single letter is shown in Table 2-1.

Amino acid name	Single letter abbreviation
Alanine	A
Arginine	R
Asparagine	N
Aspartic Acid	D
Cysteine	C
Glutamic Acid	E
Glutamine	Q
Glycine	G
Histidine	H
Isoleucine	I
Leucine	L
Lysine	K
Methionine	M
Phenylalanine	F
Proline	P
Serine	S
Threonine	T
Tryptophan	W
Tyrosine	Y
Valine	V

Table 2-1: List of amino acid names and single letter codes.

Although the 20 amino acids differ in their chemical properties, they have similar composition, with a carbon atom at the center that binds to an amino group (NH3), a carboxyl group (COO), a hydrogen atom (H), and a side chain (S). The basic structure of an amino acid is illustrated in Figure 2-3.

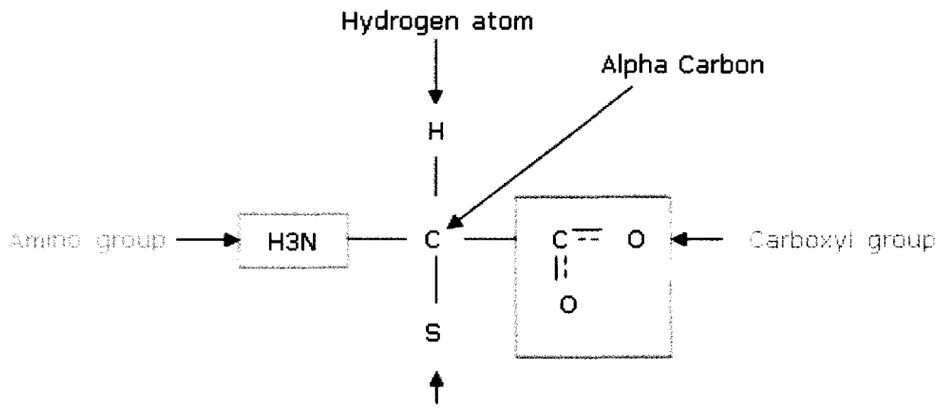

Figure 2-3 Basic structure of an amino acid.

The side chains are chemical molecules that have different properties and composition in each amino acid. References [21][22] provide a detailed illustration of structure, names, and abbreviations of 20 amino acids.

Peptide bonds are formed when the carboxyl group of an amino acid reacts with the amino group of a neighbouring amino acid. In this process, a water molecule is produced and a covalent bond is formed between the carbonyl carbon of the carboxyl group and nitrogen of the amino group. A protein chain is formed by the development and succession of peptide bonds between amino acids. A protein chain is written left to right, beginning with the amino group on the left which is referred to as N terminus, and the carboxyl group on the right which is referred to as C terminus [23].

2.1.2.2 Secondary Structure

Secondary structure refers to the folding of amino acids of the protein chain into either helices, beta-sheets, or coils. Secondary structure arises from the hydrogen bonds between

the amino group of an amino acid residue and the carbonyl group of another amino acid residue. Most of hydrophilic amino acids in proteins are facing out toward the solvent, whereas hydrophobic amino acids tend to face in toward the core of the protein, away from the surrounding solvent. This protein core is referred to as the hydrophobic core and it has a great influence on protein structure.

2.1.2.2.1Alpha Helix

The alpha helix is one of the most common types of secondary structure elements representing 32-38% of all residues [24][25]. The structure of the protein chain in an alpha helix looks like a coiled spring as illustrated in Figure 2-4. Each complete turn contains 3.6 amino acid residues. Almost all alpha helices in protein structures are right handed [26]. The formation of an alpha helix is stabilized by hydrogen bonds that occur between the carbonyl group from one amino acid and the nitrogen from another amino acid that is four residues away along the peptide chain [26]. The hydrogen bonds are approximately parallel to the helix axis [27]. This location and direction of hydrogen bonds creates a helical coiling of the backbone such that the side chains tend to point out on the outsides of the helix. The alignment and the orientation of the side chains play an important role in the determination of the protein structure and function [23].

Figure 2-4: Alpha helix reproduced from [28].

There are two other less frequent types of helix referred to as 3_{10} helix and $\pi-\text{helix}$. The hydrogen bonds occur between the carbonyl group and the NH group that are three residues away in the 3_{10} helix [26] and 5 residues away in the $\pi-\text{helix}$ [26]. The 3_{10} helix usually form the last turn of an alpha helix at the C terminus [26].

2.1.2.2.2 Beta sheets

Beta sheets are the second most common type of secondary structure accounting for 20-28% of all residues [24][25]. They are composed of 2 or more extended beta strands as shown in Figure 2-5. The alignment of beta strands to form beta sheets is stabilized by hydrogen bonds between amide nitrogen and carbonyl carbons. Beta sheets are either parallel, anti-parallel, or mixed sheets. In parallel beta sheets all strands are aligned in the same direction, with both sides buried, thus the sequences at the centre are hydrophobic and the ends are hydrophilic [26]. In anti-parallel sheets all strands run in opposite directions with one side is buried and other side is exposed to solvent, so they alternate between hydrophobic and hydrophilic [26]. Whereas in mixed sheets some strands are parallel and others are anti-parallel. The side chains in a beta strand point alternately in opposite directions along the strand and are perpendicular to the hydrogen bonds [26].

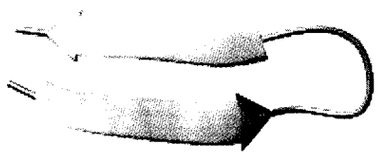

Figure 2-5: Beta sheet reproduced from reference [28].

2.1.2.2.3Random coils

Only protein conformations found in alpha helices and beta strands are composed of repeated structures of similar backbone torsion angles [26]. However, there exist protein conformations which are composed of non-repeated but of well ordered structures that are referred to as random turns or coils [29][30]. Approximately one third of residues in globular proteins are contained in turns [23].

2.1.2.3 Tertiary Structure

Tertiary structure refers to the twisting and folding of helices and sheets of the secondary structure upon themselves into a three-dimensional shape [27] as illustrated in Figure 2-6. The tertiary structure arises from the interaction between different regions of the side chains with each other by hydrogen bonds, and also arises from the fact that some chains are hydrophilic or hydrophobic and thus organize themselves accordingly in solution. The bonds that stabilize the tertiary structures of proteins can be Van der Waals bonds, ionic bonds, hydrogen bonds, or covalent bonds.

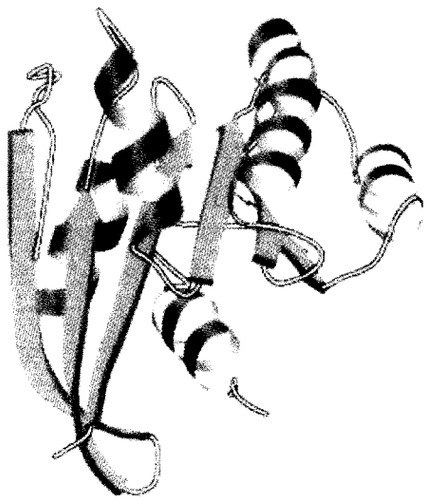

Figure 2-6: Tertiary structure of a protein composed of helices, parallel beta sheets, and coils reproduced from reference [28]. Helices are shown in red, beta strands are shown as blue arrows.

2.1.2.4 Quaternary structure

Quaternary structure is the highest level in the protein structural hierarchy. It arises from the interaction between tertiary units [23][24]. Proteins are normally functional when in their quaternary structure.

2.1.3 General classification of families of protein structures

Classification of protein structure is based on the secondary and tertiary structure [17]. Proteins may be classified as either 'all alpha-structure', where the protein core is composed

only from alpha helices, or 'all beta-structure', in which the protein is composed only from beta sheets, 'alpha+beta', where the core is composed of alpha helices and beta sheets in separated domains, as 'alpha/beta', where proteins are composed of mixed segments of alpha helices and beta sheets that interchange along the polypeptide chain [23], or as 'other' with small amounts or no secondary structure.

2.1.4 Determination of protein structures experimentally

Understanding the three-dimensional structure of a protein is one of the most important problems in bioinformatics. The three dimensional structure of a protein can be determined through complex experimental procedures. X-ray crystallography and Nuclear magnetic resonance are currently the most common methods for determining protein structures.

2.1.4.1 X-ray crystallography

An X-ray crystallography experiment is accomplished by the formation of a crystal from a protein, then exposing this crystal to X-rays. A crystal is composed of regular repeating three dimensional groups of atoms. This repeating model is employed to produce a high quality diffraction pattern (an electron density map) that reveals the crystal's molecular structure when exposed to X-rays [31]. Position and intensity of the produced pattern is examined to determine molecular size and composition. Then calculations are made in order to solve the phase interactions in the produced diffraction pattern. Afterwards specific computer software and much human intervention are used to arrive at the optimum fit between observed diffraction pattern intensities and calculations from the model structure. One of the major limitation of X-ray crystallography is to find under which conditions a given protein will have the propensity to crystallize [32]. The crystallization process is greatly affected by many parameters including the purity of proteins and other biochemical, biophysical, and biological parameters [33]. Therefore, the variation and dependencies of these parameters can affect the crystallization output [33]. So, the most difficult step in X-

ray crystallography is to find the optimal parameter set under which to create a crystal. X-ray crystallography requires significant expertise and time on the part of the scientist.

2.1.4.2 Nuclear magnetic resonance spectroscopy

Nuclear magnetic resonance (NMR) spectroscopy is one of the powerful experimental techniques for determination of three dimensional structures of proteins. NMR is based on the idea that proteins perform their physiological functions in body fluids such as blood, stomach liquid, and saliva. Accordingly, solutions used in NMR are adjusted to mimic biological fluids [34]. NMR experiments have the advantage that data is recorded in solution and does not require crystallization of proteins. NMR is accomplished by exposing molecules of a protein in a solution to nuclear magnetic resonance which result in molecular vibration and movement. Best results of NMR experiments are achieved when molecules vibrate extremely fast, and this limits the size of molecules to only small ones (approximately 30 kD [32]), thereby limiting the length of protein chain whose structure may be determined via NMR. NMR measures the movement of atomic nuclei, and these movements are greatly influenced by distances of nearby atoms.

The output of NMR is a collection of distances between pairs of atoms, and represents a group of models, instead of a single structure. The average position of each atom in these models is calculated. The average model is then adjusted to follow normal bond distances and angles [32]. The result of NMR experiments are less accurate than those obtained by x-ray crystallography [32][35]. The major limitation of NMR experiments is that only small proteins can be investigated [36][37]. This limitation could be reduced through recent advances in NMR methods and new biochemical processes [37]. In ref [38], researchers have succeeded to investigate large molecules of size about ten times the traditional size limit by using transverse relaxation optimized spectroscopy and cross-correlated relaxation enhanced polarization transfer techniques.

2.1.5 Prediction of three-dimensional structure of proteins

The success of genome sequencing projects has resulted in influx of sequence data. Although experimental determination of three-dimensional structure of proteins provides high resolution structural models, they can only be applied to a subset of proteins, and each experiment requires great amounts of time, expertise, and resources. As a result, computational three-dimensional structure prediction has attracted many researchers. There are two main approaches to predict the three-dimensional structure of proteins. The first approach includes homology and threading methods which necessitate the presence of a homologous protein structure that share high degree of sequence similarity to the protein sequence to be modeled. The second approach includes the *ab-initio* method that predicts protein three-dimensional structure without relying on similarities between the protein sequence to be modeled and any of the known structures.

2.1.5.1 Homology modeling

Homology modeling is based on two main concepts. The first concept is that protein three-dimensional structure is uniquely determined by its amino acid sequence [40][41], accordingly knowing the protein sequence would suffice to determine its structure [42]. The second concept is that protein structure is more conserved than protein sequence [43][44], so proteins with similar sequences adopt similar structures [46][47]. This fact was quantified by Rost in 1991, as he found that proteins which share more than 35% of identical residues are structurally similar [48][49].

Homology modeling involves a multistep procedure. The first step is to recognise and find templates of known structures that share similar sequences to the protein sequence to be modeled. The templates for modeling can be obtained by sequence comparison methods, for example BLAST [50] or PSIBLAST [51]. If the pairwise sequence identity between the template and the target to be modeled is low, multiple sequence alignment from

homologous proteins could be used to solve this problem [42]. The second step is to generate the backbone by simply copying the coordinates of the template back bone to the model. If the aligned residues in both template and target differ, only the backbone coordinates are copied, else if the residues are the same, the side chains could also be copied from the template as this yields better accuracy than predicting the side chain conformations [42]. The third step is loop modeling which is performed by either following a knowledge-based approach or an energy-based approach [42]. The knowledge-based approach is performed by searching the PDB for known loops with end points similar to the residues between which the loop will be placed and then copy the matching loop. The energy-based approach is performed by using an energy function to assess the loop quality. This energy function is then minimized, for example, by using Monte Carlo approaches [52], or using molecular dynamics [53][54] to obtain the best loop conformation. After that, the model is refined to correct stereochemical and geometrical errors at places where regions of the protein chain differ significantly from the template protein. Then the last step is to validate the model which is done by either testing whether the bond lengths and bond angles are within standard limits, or by evaluating stereochemical characteristics and inter-atomic distances. The quality of the homology modeled structure is greatly affected by the degree of similarity existing between the target and the model sequences. If the sequence similarity between the target and the model is higher than 50%, then the modeled structure is likely to share more than 90% of the target structure [55]. Errors in homology modeling are mainly due to incorrect alignment of target and template. Other errors include distortions of back bone, loops, and side chains [56]. Recently, homology modeling has been greatly improved due to the growth in the database of known protein structures and due to enhancements in modeling software [56].

2.1.5.2 Threading

In fold recognition by threading, fold assignment is achieved by threading an amino acid sequence through each of the structures in a library of all known three dimensional protein structures [57]. After that the quality of each sequence-structure alignment is assessed to determine which models are most likely to be correct [57]. It is computationally very expensive to perform this assessment step using a detailed energy evaluation for every possible model [58]. Thus, approximate energy parameters are employed to evaluate the interactions of amino acid residues instead of every atom in the protein [58]. Recently, there have been many trials to employ neural networks to determine different parameters that can be used in order to facilitate the threading techniques [59][60].

2.1.5.3 *Ab initio* approach

Homology and threading methods are very successful approaches for protein structure prediction, but they necessitate the availability of homologous proteins in the databases. Thus, *ab initio* or *de novo* prediction methods have been developed which are based on the thermodynamic hypothesis [61] which states that the native structure of a protein is the one that corresponds to the global minimum of its free energy. The a*b initio* approach is achieved by modelling all the potential energy functions involved in the process of protein folding in order to search for a protein conformation with lowest free energy. This search is typically achieved by using energy minimization approaches, for example, simulated annealing, molecular dynamics, genetic algorithms, or Monte Carlo minimization. *Ab initio* approaches based only on the thermodynamic hypothesis are considered unfeasible [62][63] because of the inaccuracy of the energy functions used to distinguish between correct from incorrect structures [64], and the lack of powerful global optimization methods for exploring the conformational space represented by those functions [65]. Thus multiple sequence alignment and results from secondary structure prediction methods are recently introduced as constraints to a*b initio* methods [63]. A*b initio* structure prediction is a very

complex procedure [68] and is extremely resource intensive [66]. Furthermore, its application is limited to relatively short polypeptides [67].

2.1.6 Importance of protein secondary structure prediction

As shown previously, the direct prediction of protein tertiary structure is a very challenging problem. Therefore, several groups have used the prediction of protein secondary structure as a first step to elucidating the three dimensional structure and function of proteins. For example, in ref [71], three-dimensional structures were predicted using torsion angle dynamics and predicted secondary structure states. Also in ref [72], three dimensional folding of protein structure was developed starting from its secondary structure. Furthermore, secondary structure predictions can be used in fold recognition approaches [70][72]. Also, secondary structure can be employed to predict aspects of protein function [69][73]. Moreover, the protein secondary structure reveals important information about distantly related proteins and conserved regions of protein sequences that are either functionally important or involved in maintaining the structure [74].

2.1.7 Using neural networks to predict protein secondary structure

Many researchers have developed secondary structure prediction methods using a wide range of approaches including support vector machines (SVMs) [2], methods based on information theory techniques (for example GOR method [3][4]), neural networks [5][6][7][8], and nearest neighbour methods[9]. In this thesis, only neural networks will be discussed as they are the most widely used technique and they achieved very high prediction accuracies [10][11].

As discussed below, the first attempt to use neural networks to predict the protein secondary structure was in the late 1980's, by two independent research groups: Qian and Sejnowski[75], and Holley and Karplus[76]. They achieved an approximate prediction

accuracy of 63% for the three secondary structure states: helix, sheet, and coil. In the 1990s, evolutionary information taken from multiple sequence alignments of related but evolutionarily diverged proteins in the same structural family was used as input to neural networks, leading to an improvement in prediction accuracies above 70% [1]. The present state of the art in neural network prediction of secondary structure is approximately 80% [1].

2.2 Neural networks

For the last few decades, scientists have tried to model and mimic the human brain capabilities to build artificial computational systems that can process information and make decisions in a similar way to human brain. The human brain is composed primary of specialised cells called neurons. Each neuron is connected to thousands of other neurons to form a highly complex network. A biological neuron consists of a cell body, an axon and dendrites as illustrated in Figure 2-7. The axon originates from the cell body and acts as an output channel. The dendrites are the input receptors of information from other neurons; these dendrites are connected to other neurons through synaptic endings. Each neuron receives signals from other neurons, these signals are either excitatory, that tend to generate other signals, or inhibitory, dampening the probability of a new signal being triggered. Every neuron will then add all received excitatory and inhibitory signals from other neurons. If the total of all received signals exceeds a specific threshold, the neuron will fire. The firing neuron will transmit a new signal to other neurons. But if the total is lower than the threshold, no signals will be transmitted to other neurons. Motivated by the idea of the neurons, scientists have tried to build artificial neurons analogous in functionality to biological neurons.

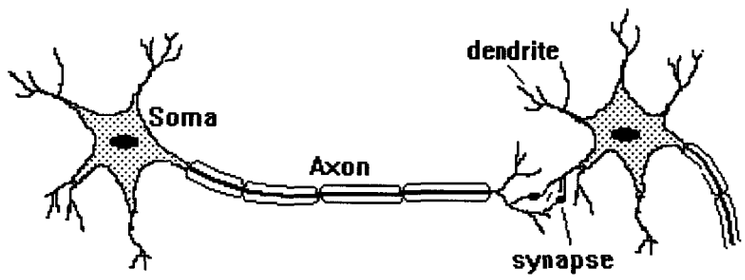

Figure 2-7 Biological neuron reproduced from reference [77].

2.2.1 Earliest attempts of neural networks

Researchers attempted in the 1940s to study neural networks to solve complex problems. The McCulloh and Pitts model, the Perceptron network, and the Adaline model were the most successful earliest attempts to construct simple neural networks.

2.2.1.1 The McCulloh and Pitts model

Two scientists, Warren McCulloh and Walter Pitts, had believed that modelling brain functionality could be represented mathematically [78][79]. The McCulloh and Pitts model was the first attempt to imitate the computing process of biological neurons. McCulloh and Pitts stated that artificial neurons can be built to transmit signals when its inputs exceed certain threshold level corresponding to the biological neuron in firing or not firing. They also stated that dendrites linking biological neurons can be imitated through interconnections. Weights can be assigned to the interconnections analogous to neuron synapses. Also excitatory and inhibitory signals can be imitated by programming the connections to be either positive or negative [80].

Although the artificial neuron model was quite simple and has limited capabilities, it however was the sparkle and backbone of many of the recent neural network architectures.

As shown in Figure 2-8, the model collects input signals m_1, m_2 , ... , m_n, multiplies them by corresponding weights w_1, w_2 , ... ,w_n, then compare the weighted sum with a threshold. The result is then applied to an activation function *a*, where the activation function used by McCulloh and Pitts is the unit step function. If the weighted sum is higher than a given threshold, the model assumes that the output is one, otherwise it is zero. The output is calculated as follows :

$$Output = a\left(\sum_{i=1}^{n} m_i w_i - \theta\right) \qquad \text{equation (2-1)}$$

where *a* is activation function, m_1, m_2, ..., m_n represent the inputs , w_1, w_2,...., w_n represent the weights, and θ is the bias.

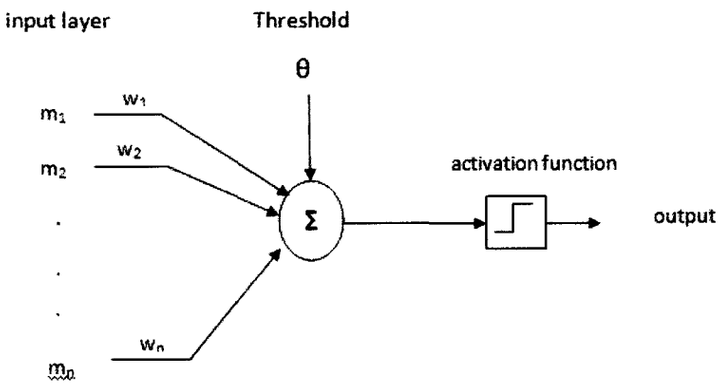

Figure 2-8: MuCulloh Pitts model.

One major limitation of this model was using fixed weights and thresholds which did not allow learning to take place.

2.2.1.2 Perceptron Network

McCulloh and Pitts model had a very large shortcoming in terms of lack of learning capabilities. New methods were required to allow the weights to be adjusted in order for the

network to work efficiently. The fixed weights in the McCulloh and Pitts model also affected the introduction of new patterns to the model, as the whole setup had to be disrupted when introducing new patterns. To overcome these major disadvantages, Frank Rosenblatt, in 1960, invented the Perceptron, which was the first machine with learning capabilities. The Perceptron was capable of pattern classification of linearly separable sets [81]. The Perceptron consists of three layers; an input layer that accepts input signals, a second layer called a feature detector unit which is composed of nodes connected to the input layer, and an output layer composed of an output node with adjustable weights. The architecture of the Perceptron is illustrated in Figure 2-9.

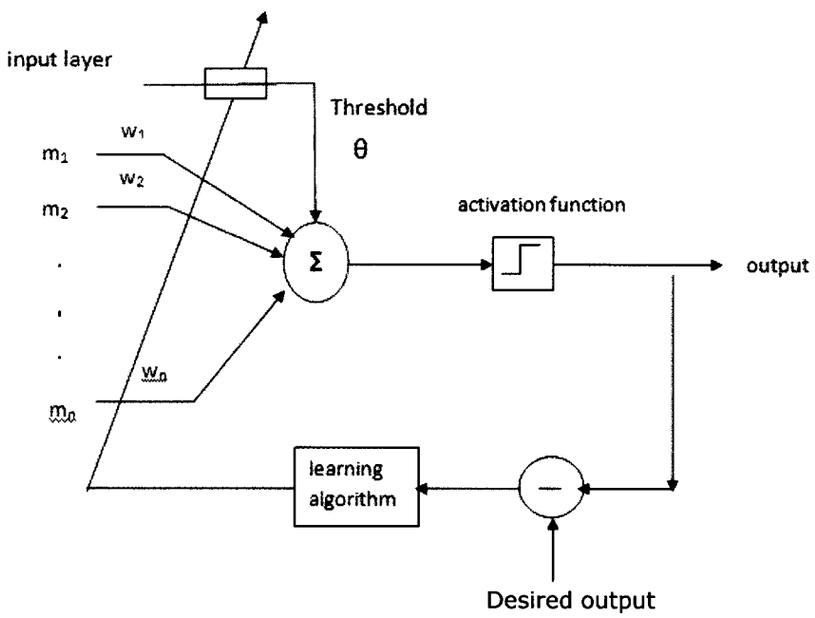

Figure 2-9: The Perceptron network.

The output is calculated as shown in equation (2-1), then the weights are updated in response to the difference between the actual output and the desired output as shown in equation (2-3).

$$\Delta w = \zeta\left(d - a\left(\sum_{i=1}^{n} m_i\, w_i - \theta\right)\right) m_i. \qquad \text{equation (2-2)}$$

where Δw is the change in weight connection, ζ is the learning rate that ranges from zero to one, d is the desired output, a is the activation function, m_i represents the input signals, w_i represents the connection weight, and θ is the bias.

Rosenblatt stated that if a set of patterns is linearly separable, then the learning algorithm will definitely find the appropriate set of weights and converge in finite number of steps [81][82].

Minsky and Papert in 1969 published a book critiquing the Perceptron of being incapable of solving non linearly separable patterns [80][83]. Non linear separable patterns are those patterns which could not be separated by single lines or hyperplanes. Minsky and Papert proved that the Perceptron was not able to solve the XOR problem, which is an example of non linearly separable functions. The powerful criticism by Minsky and Papert led to a decline of neural network research for a number of years [80].

2.2.1.3 Adaline and Madaline

Based on the idea of the McCulloh and Pitts model, Bernard Widrow in 1962, developed another neural network model that was called Adaline (Adaptive linear neuron) [85]. Widrow's model was composed of one layer with a more sophisticated learning algorithm than the Perceptron. Widrow and Hoff in 1960, developed the least mean square rule (LMS), which in generalized form became the learning rule underlying the backpropagation rule which is the most widely used in different network architectures.

The Adaline model used a hard limiter activation function, and the weights were adjusted according to the LMS rule, where they were incremented in every iteration by an amount proportional to the total error of the network [85]. The weights are updated according to the following equation:

$$\Delta w = -\zeta \nabla E(w). \qquad \text{equation (2-3)}$$

where Δw is the change in weight , ζ is the learning rate that ranges from 0 to 1, and *E(w)* is the cumulative error of the network. *E(w)* is calculated as the square of the sum of the difference between the desired output *d*, and the actual output. The actual output is the difference between the weighted sum of the inputs and the bias θ is calculated as follows:

$$\text{Output} = (\Sigma_i w_i m_i^k - \theta). \qquad \text{equation (2-4)}$$

So the equation that calculates the cumulative error for k patterns of inputs is:

$$E(w) = \sum_{k=1}^{z} \left(\left(d(k) - \left(\sum_{i=1}^{n} w_i m_i - \theta \right) \right) \right)^2. \qquad \text{equation (2-5)}$$

The Adaline model is trained in a similar way to the training of the Perceptron except the weights are updated using the difference between the desired output and the actual output before going to the activation function as shown in Figure 2-10 [85].

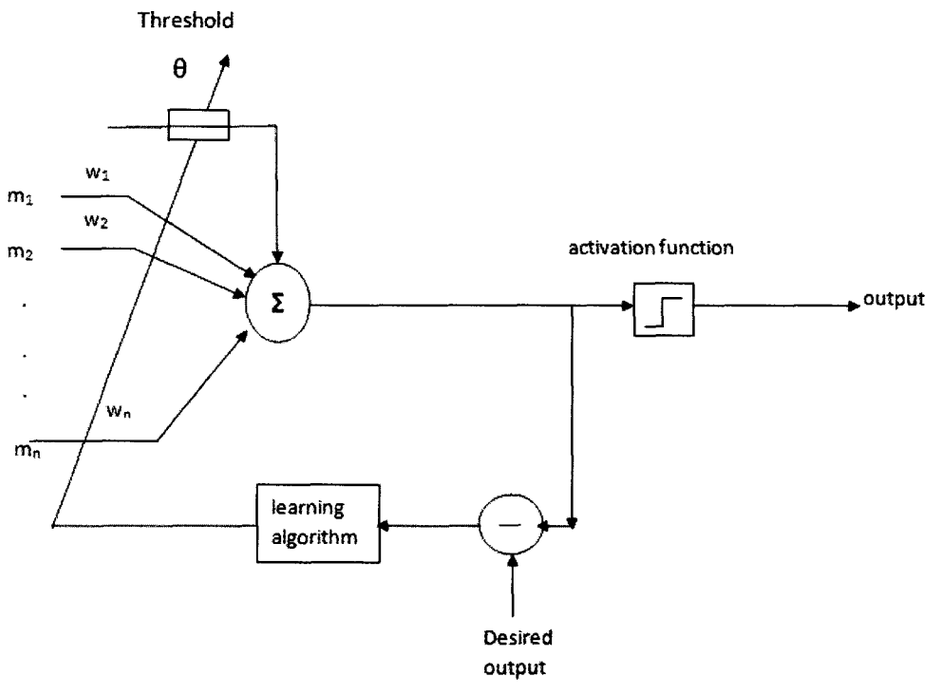

Figure 2-10: The Adaline model.

Although the Adaline had very powerful training capabilities and had a better generalization capabilities compared to the Perceptron, it still was incapable of solving non linear separable patterns. So two years later after the Adaline, Widrow developed the Madaline which is combination of a number of Adaline units.

The Madaline was capable of solving the XOR problem [85]. In the same year Widrow developed the first application to utilise neural networks in real life applications, such as meteorological forecasts, character recognition, and speech recognition [85].

2.2.2 Learning algorithms in neural networks

Neurons that make up a neural network are connected in such a way to signal each other as information is processed. Every connection is assigned a connection weight that differs in

magnitude from other weights in the model. Training a neural network starts by the assignment of random weights to the weight matrix. The weights are then adjusted according to the performance of the network. The weight adjustment is repeated several times until the error between the desired and actual output is within a reasonably small range.

Training a neural network falls into three main categories: supervised, unsupervised and reinforcement training [84][85].

2.2.2.1 Supervised learning

Supervised learning is accomplished by presenting a set of known inputs and their associated outputs. The neural network is taken through an optimization process in an attempt to minimize the cumulative sum of errors between the actual and the desired output when presented with the training input data. The weights are adjusted through several iterations according to a given training mechanism. As more input and output patterns are presented to the model, the updating process of the weights continues in order to adapt to new patterns. The back propagation [86] and the least mean square algorithms [87] are the most widely used supervised rules [85].

2.2.2.2 Unsupervised learning

Unsupervised learning differs from the supervised learning in that there are no outputs provided to the network during training. There are several network architectures and training algorithms which can be used for unsupervised learning. For example, with Kohonen Self Organizing Maps (KSOM), unsupervised learning occurs when the Kohonen network has to classify input patterns into different categories [85]. When the input patterns are applied to the network, the weights are adjusted according to competition between the output nodes. The winning node is the node with the highest score. After that

the unsupervised algorithm strengthens the weights between the input and the winning node, and also updates the weights of the nodes in the neighbourhood of the winning node. The training process proceeds until the neural network organizes the data into different classification groups.

2.2.2.3 Reinforcement learning

In reinforcement learning , for each input the neural network is told whether the output is right or wrong. If the output is correct, the weights leading to that output are reinforced, otherwise they are weakened. Reinforcement learning differs from supervised and unsupervised learning. It differs from the supervised learning that the data provided to the network does not contain any anticipated outputs, and differs from the unsupervised learning as in unsupervised learning, the neural network is not told if the output is correct.

2.2.3 Neural network architectures

Neural networks are composed of neurons which are grouped into a number of layers. There are three types of layers: input, output, and hidden layers. A schematic representation of a neural network composed of an input layer, 2 hidden layers, and an output layer is illustrated in Figure 2-11. The input layer is composed of neurons that receive input patterns from the external environment. The output layer is composed of a number of output nodes that present the output of the network to the external environment. The number of output neurons is related to the intended use of the neural network. For example, if the neural network is assigned to classify items into a number of categories, then there should be one output neuron for each category.

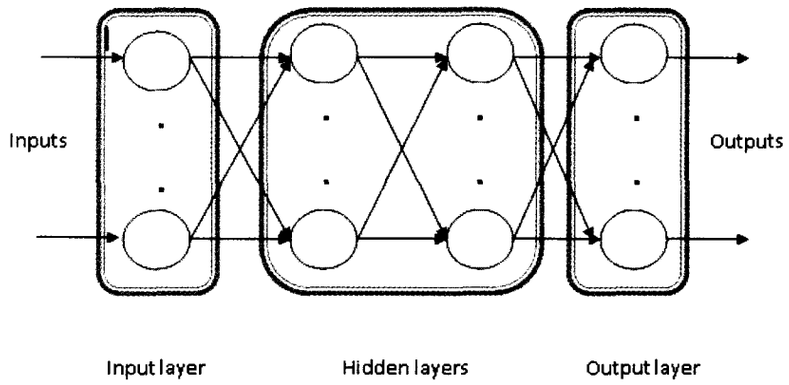

Figure 2-11: A schematic representation of a neural network.

Hidden layers are the layers between the input and output layers. The hidden layers contain most of the computational power of neural networks [85]. The utilisation of one hidden layer is the most wide spread method for many practical problems, as it can approximate any function that contains a continuous mapping between two finite spaces [84]. Although there are not many problems that require using two hidden layers, nevertheless the use of two hidden layers has the capability representing functions with arbitrary shape [84].

Some researchers have classified neural network architectures according to the organization of nodes and the way the data is processed through the network into two main topologies: Feed forward and recurrent architectures[85]. These topologies will be discussed in more details in chapter 4, but are briefly introduced below.

2.2.3.1 Feed forward neural networks

Feed forward neural networks are composed of hierarchically organized layers. It starts with an input layer that is connected to a hidden layer. This hidden layer is either connected to another hidden layer or an output layer. Every neuron in one layer is connected to each neuron on the next layer and there is no connection between neurons in the same layer. So

the data is fed forward between the layers; information enters at the inputs and passes through the network in a unidirectional path, layer by layer, until it arrives at the output. There is no feedback between layers and this is why they are called feed forward neural networks.

2.2.3.2 Recurrent neural networks

Recurrent neural networks are networks with feedback connections. They are capable of learning time-varying patterns, where the output of a certain input pattern is dependent on the previous states [88]. Recurrent neural networks have been used to solve a wide variety of problems. In ref [89], a new type of recurrent neural network architecture was proposed in which each output unit is connected with itself and is also fully-connected with other output units and all hidden units. The proposed recurrent neural network has shown improved performance in the generalization power compared to multilayer feed forward neural networks and other recurrent networks [89].

2.2.4 Training neural networks

In this thesis, we have used five different neural network architectures to cover static, dynamic, historical, and radial basis networks. Static networks are characterized by the presence of feed-forward connections with no delays and no feedback elements. An example of static networks are simple feed-forward networks. While the dynamic networks are characterized by the dependence of the output on previous inputs, outputs, or states of the network. Dynamic networks generally contain feedback, or recurrent, connections. Dynamic networks often outperform the static networks due to their capability of learning time-varying patterns and also due to the presence of memory [133]. This has led to the use of dynamic networks in many different fields as shown in references [134-137]. Examples of dynamic networks are the recurrent neural networks and nonlinear autoregressive networks with exogenous inputs. Historical networks are partial recurrent networks; two popular

examples of these networks are Elman and Hopfield networks. Radial basis networks require more nodes than feed forward networks, however they can be trained in a fraction of the time it takes to train feed-forward networks. The most popular examples of these networks are radial basis networks [138], probabilistic neural networks [139], and generalized regression networks [140].

A quick review of feed-forward networks, recurrent neural networks, nonlinear autoregressive networks with exogenous inputs, Elman networks, and probabilistic neural networks will be discussed generally following references [79][133][141].

2.2.4.1 Feed forward neural networks

The feed-forward network was invented in the 1960s to solve the limitation of the Perceptron in solving non linear separable spaces. However, there was no efficient leaning algorithm at that time that was capable of training this model. In 1974, Werbos proposed a new leaning algorithm called backpropagation learning [80], that was used for feed-forward networks with great success. The feedforward network will be disussed below, but details of the backpropagation algorithm are left to Appendix A.

Topology of the feed-forward network

The feed forward term illustrates the way the network processes and recalls patterns. The neurons are only connected in a forward direction, and each layer is connected to the subsequent layer with no connections in the backward direction. The multilayer Perceptron (feed-forward network) is composed of an input layer, one or more hidden layers, and an output layer. Each neuron in the input layer is connected to all the neurons in the hidden layer. Also each neuron in the hidden layer is connected to all the neurons in the subsequent layer which is either another hidden layer or the output layer.

Choosing the number of hidden layers and the number of neurons in these hidden layers has a direct impact on the performance of the feed forward neural networks[142]. A schematic representation of the multilayer feed-forward network that is composed of an input layer, one hidden layer and an output layer is shown in Figure 2-12.

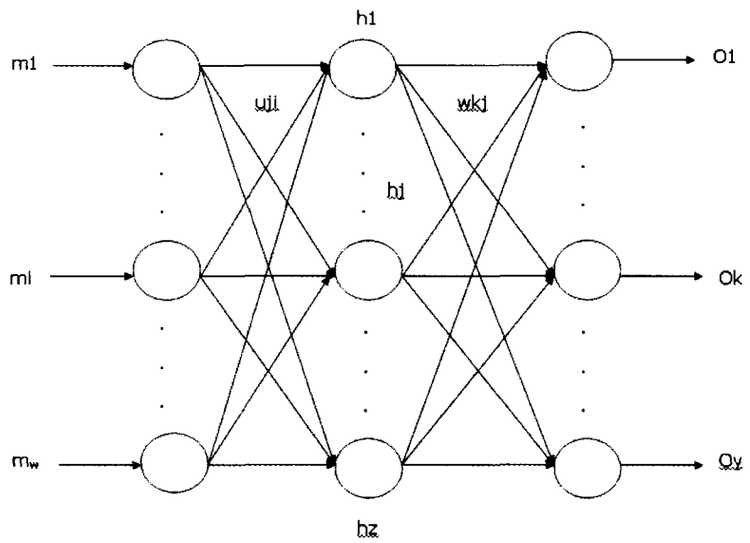

Figure 2-12: A schematic representation of feed forward networks.

Here, $m_1...m_w$ are the inputs, the index i represents the i^{th} neuron in the input layer that is composed of w neurons. $h_1...h_z$ are the hidden neurons, the index j represents the j^{th} neuron in the hidden layer that is composed of z neurons. $O_1...O_y$ are the outputs, where the index k represents the k^{th} neuron in the output layer that is composed of y neurons. u_{ji} represents the connection weight between the i^{th} neuron in the input layer and the j^{th} neuron in the hidden layer. w_{kj} represents the connection weight between the j^{th} neuron in the hidden layer and the k^{th} neuron in the output layer.

Each node represents a simple Perceptron, that computes the weighted sum of all inputs to that node and then apply this sum to an activation function. A schematic representation of each node is shown in Figure 2-13.

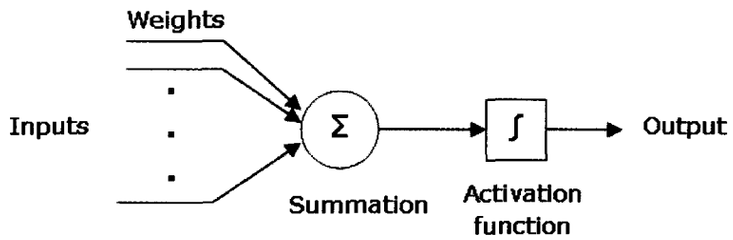

Figure 2-13: A schematic representation of a node in the multilayer Perceptron.

Computing the output of feed-forward neural networks

The output of the hidden layer h_j in a feedforward network is expressed by:

$$h_j = a(hin_j) \qquad \text{equation (2-6)}$$

where (a) is the activation function and hin_j is the weighted sum of the inputs to node h_j which is equal to:

$$h_{inj} = \sum_{i=1}^{w} u_{ji} m_i . \qquad \text{equation (4-2)}$$

So the output of the hidden layer is:

$$h_j = a\left(\sum_{i=1}^{w} u_{ji} m_i \right). \qquad \text{equation (2-7)}$$

Similarly, the output (O_k)of the k^{th} neuron in the output layer is equal to :

$$O_k = a(Oin_k) \qquad \text{equation (2-8)}$$

where (a) is the activation function, and O_{ink} is the weighted sum of the inputs to node O_k which is equal to :

$$O_{ink} = \sum_{j=1}^{z} w_{kj} h_j . \qquad \text{equation (2-9)}$$

So the output is :

$$O_k = a\left(\sum_{j=1}^{z} w_{kj} h_j \right) \qquad \text{equation (2-10)}$$

By substituting the value of h_j from equation (4-3), the output becomes:

$$O_k = a\left(\sum_{j=1}^{z} w_{kj}\, a\left(\sum_{i=1}^{w} u_{ji} m_i\right)\right). \qquad \text{equation (2-11)}$$

Training of feedforward neural networks

Feedforward networks may be trained using a number of learning algorithms. Backpropagation [80] is used in this thesis. A detailed description of backpropagation is given in Appendix A of the thesis.

In this thesis, MATLAB neural network tool box was used to create feed-forward backpropagation networks using the command (newff).

2.2.4.2 Recurrent neural networks

The recurrent neural network (RNN) is the most popular type of dynamic network. The weights of dynamic networks affect the network output directly and indirectly. They affect the output directly as any alteration in the weight results in instant alteration in the output at the current time step. This direct effect can be calculated with standard backpropagation. They affect the network output indirectly as some of the inputs to the output layer are calculated from previous time step. This indirect effect can be calculated with the computationally intensive dynamic backpropagation [144][145].

The RNN has feedback connections from nodes in the output layer to some nodes in the input layer and some self-loops in the hidden layer. The architecture of RNN is shown in Figure 2-14. The feedback loop has a unit time step delay in all the layers of the network except the last layer [118].

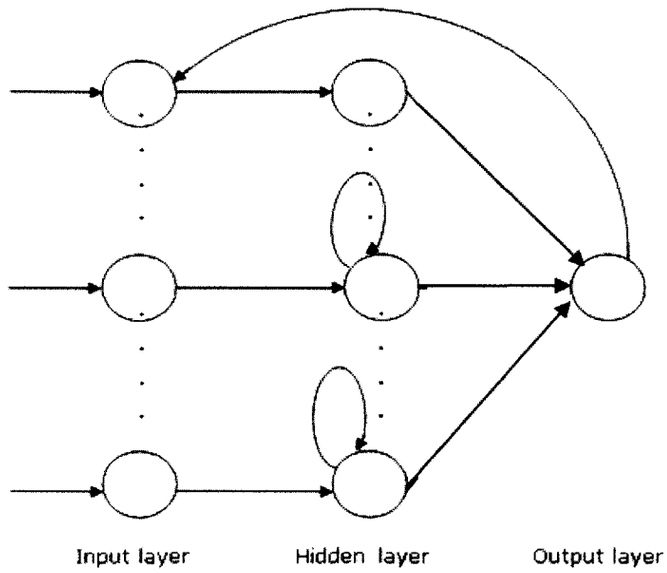

Figure 2-14: A schematic representation of the RNN.

The output of any node in the RNN is a function of the current and previous inputs to that node. The output can be calculated as follows:

$$O_j(t) = a\left(net_j(t-1)\right) = a\left(\sum_i w_{ji} O_i(t-1) + x_j(t-1)\right) \qquad \text{equation (2-12)}$$

Where $O_j(t)$ is the output of node j at time t, a is the activation function, net_j is the net input to node j at previous time step (t-1), w_{ji} is the connection weight between node j and node i, and $O_i(t-1)$ is the output of node i at previous time step(t-1).

In this thesis, MATLAB neural network tool box was used to create RNN using the command (newlrn). The RNN is trained using gradient-based algorithms.

2.2.4.3 Nonlinear autoregressive network with exogenous inputs (NARX)

The NARX is a recurrent dynamic network, with feedback connections between previous input and output nodes and the current output nodes. The NARX model is based on the autoregressive exogenous model (ARX), which is one of the quantitative dynamics modeling approaches which have been used frequently in time-series modeling [146].

The output signal for the NARX model depends on both values of the output signal at previous state(s) and also on values of the exogenous input signal at a previous state [118]. The output signal *o(t)* is calculated as follows:

$$o(t) = f\left(o(t-1), o(t-2), \ldots . o(t-n_x), x(t-1), x(t-2), \ldots, x(t-n_z)\right) \quad \text{equation (2-13)}$$

The NARX model can be created using a feed-forward neural network to approximate the function *f*. There are two main architectures to implement a NARX model: parallel architecture and series-parallel architecture; these two architectures are shown in Figure 2-15, where o(t) represents the calculated output, O'(t) represents the true ouput, and x(t) represents the input. The calculated output of the NARX model of the parallel architecture is fed back to the input of the feed-forward neural network. While the true output of the NARX model of the series-parallel architecture is fed back to the input of the feed-forward neural network during training instead of feeding back the estimated output. The series-parallel architecture has the advantage of using a more accurate input feed-forward network as it is using the true output.

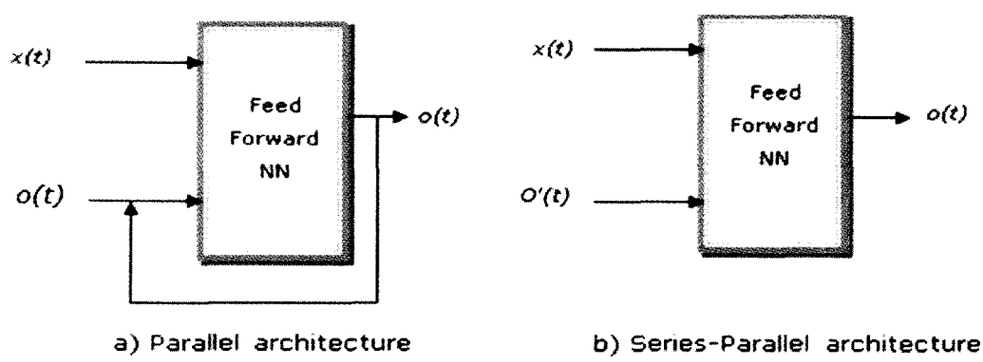

Figure 2-15: A schematic representation of the NARX a) shows the parallel architecture b) shows the series-parallel architecture.

In this thesis, MATLAB neural network tool box was used to create NARX networks using the command (newnarx).

2.2.4.4 Elman networks

An Elman network is a kind of historical neural network that was proposed by Elman in 1990 [147]. It is composed of an input layer, hidden layer, context layer, and an output layer. The context layer memorizes the output of the previous state of the hidden layer, so it produces a unit-step time delay [148][149]. Elman networks combine some characteristics of both feed-forward and recurrent neural networks. Elman networks resemble feed-forward networks in that they contain a feed-forward loop that is composed of input, hidden, and output layers connected by variable weights, and it resembles recurrent networks with a back-forward loop that is composed of hidden and context layers connected by fixed weights [148]. The back-forward loop increases the dynamic capability of Elman networks and increase its adapting capability to time varying patterns [149]. This has led to the use of Elman networks widely in the field of dynamic system identification [148]. The architecture of Elman network is shown in Figure 2-16.

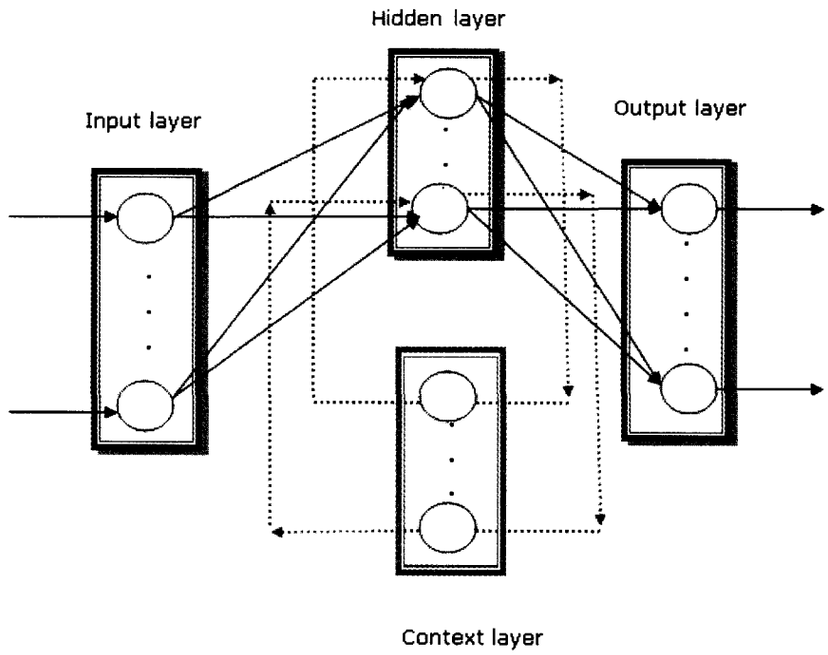

Figure 2-16: A schematic representation of Elman network.

The input layer is composed of n nodes, the output layer is composed of k nodes, and each of the hidden layer and the context layer are composed of z nodes. The weights connecting the input layer and the hidden layer are denoted by *w1*, the weights connecting the hidden layer and the context layer are denoted by *w2*, and the weights connecting the hidden layer and the output layer are denoted by *w3*. The inputs of Elman network are denoted by *x(t-1)*, the outputs of the hidden layer are denoted by *h(t)*, the outputs of the context layer are denoted by *c(t)*, and the outputs of Elman network are denoted by *O(t)*.

The output of the hidden layer *h(t)* is calculated as follows:

$$h(t) = g\big(w1x(t-1) + w2c(t)\big) \qquad \text{equation (2-14)}$$

The output of the context layer *c(t)* is calculated as follows:

$$c(t) = h(t-1) \qquad \text{equation (2-15)}$$

The output of the output layer *O(t)* is calculated as follows:

$$O(t) = f\big(w3h(t)\big) \qquad \text{equation (2-16)}$$

where *g* is a nonlinear activation function in the hidden layer nodes which often takes the form of a sigmoid function [147] which is equal to:

$$g(s) = \frac{1}{1+e^{-s}} \qquad \text{equation (2-17)}$$

and *f* is a linear function [147].

The error *E* of the network is calculated as follows:

$$E = \sum(T_t - O_t)^2 \qquad \text{equation (2-18)}$$

where *T* represents the target vectors.

Elman networks are trained using back propagation with momentum to update the weights and biases in order to reduce the error of the network [150]. In this thesis, MATLAB neural network tool box was used to create Elman network using the function (newelm). The

weights and biases of each layer are initialized with the Nguyen-Widrow layer-initialization method [151].

2.2.4.5 Probabilistic neural networks

Probabilistic neural networks (PNNs) are neural network implementation of Parzen windows [152]. They were originally created and used due to their extremely fast training time on real life problems [153]. PNNs were first used by Specht in 1990 [154]. They are composed of an input layer, a hidden layer, and an output layer. As shown in Figure 2-17, the input layer is composed of input nodes, the hidden layer is composed of pattern units, and the output layer is composed of category units. Every input unit is connected to all of the pattern units of the hidden layer, and every pattern unit is connected to only one of the category units.

In order for the PNN to classify a new test vector, the test vector is applied to the network. The input layer calculates the distances between the test vector and the training inputs. Then the input layer produces a vector which determines how close the test vector is to the training inputs [155]. The second layer sums these contributions from all input units and produces a vector of probabilities [155]. On the output of the second layer, there is a transfer function which picks the maximum of these probabilities, and produces one for the class of maximum probability and zeros for the other classes [155].

The main advantage of the PNNs is the training speed which is extremely fast relative to other network architectures [153][156]. Also PNNs are relatively insensitive to outliers [157]. In addition, new training patterns can be added into a previously trained PNN without any problem [156][158]. The main limitation of PNNs is in the storage; they need large memory space to store the model since there is one hidden node per training sample [159].

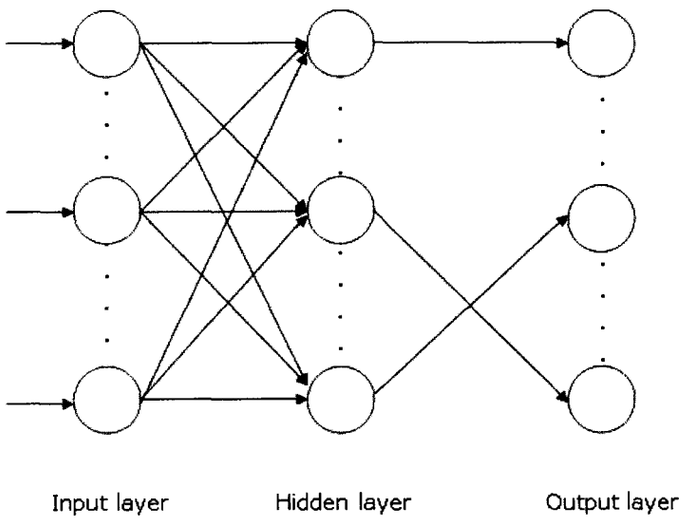

Figure 2-17: A schematic representation of a PNN.

In this thesis, MATLAB neural network tool box was used to create PNN using the command (newpnn).

2.2.5 Earlier attempts of using neural networks to predict protein secondary structure

In 1987, Qian and Sejnowski [75] described a feed forward neural network that was based on a previous application of speech synthesis [90]. Qian and Sejnowski replaced the input string of letters with amino acids and the output phonetic symbols with secondary structure types. They used a list of 106 proteins obtained from the Brookhaven National Laboratory. A subset of these proteins was taken for training and the remaining proteins were used for testing. The test set that was non-homologous with the training set, meaning that no significant sequence similarity existed between any pair of training and testing proteins. Their model was composed of an input layer, one hidden layer, and an output layer. The Input layer examined a segment of primary amino acid sequence of width 13-17 residues, with each residue represented by 21 individual units representing each of possible amino acid. The hidden layer was composed of 40 neurons [75]. The output layer was composed of

3 neurons that corresponded to the likelihood that the central residue is an α-helix, β-sheet, or coil. The network outputs were compared to the expected conformational states. Errors were used to adjust the weights in the model through the backpropagation algorithm. The average success rate of Qian and Sejnowski model was 64.3% on the three states of secondary structure (alpha-helix, beta-sheet, and coil)[75]. The performance of Qian and Sejnowski model was compared against three of the most commonly used methods at that time (GOR [3], Chou and Fasman [91], and Lim [92]) and it was found that Qian and Sejnowski model outperformed those methods [75].

In 1989, another research group, Holley and Karplus [76], independently used a neural network very similar to the one used by Qian and Sejnowski. The input layer was composed of 17 residues, each residue was represented by 21 input units. The hidden layer was composed only of two units. The output layer was also composed of two units. Secondary structure in these output units was encoded such that (1,0) represented helix, (0,1) represented sheet, and (0,0) represented coil. The method achieved an approximate predictive accuracy of 63% for the three secondary structure states: helix, sheet, and coil.

Both of these early approaches, achieved limited accuracy since the networks were not provided with any information regarding non-local interaction between distant pairs of amino acids. As discussed below, this was remedied with the use of evolutionary profiles as input data rather than simple sequence windows.

2.2.6 Popular neural networks used to predict protein secondary structure

2.2.6.1 PHD

Rost and Sander developed a much more elaborate system to predict protein secondary structure that is called PHD [93]. The PHD model is composed of three levels of networks.

The first level is a sequence-to-structure level. The input to the first level is a window of 13 residues, each represented by 21 units, and the output represents the secondary structure state of the central residue as either helix, strand, or coil. The outputs from the first level are then fed into a second level. The second level is a structure-to-structure level that maps secondary structure information from its inputs to secondary structure information at its output. The third level of PHD computes the arithmetic average (jury decision) between a number of independently trained networks.

A new aspect was applied in encoding the input signals, where evolutionary information was provided in the form of multiple sequence alignments instead of using single sequences. Rost and Sander used 130 protein chains for training the PHD. A 7-fold cross validation was used to test the performance of the PHD, where 111 protein chains were used for training and 19 for testing. Then this was repeated 7 times until all proteins had been used for testing one time [93]. By using three network levels and using multiple sequence alignments to encode the input signals, the prediction accuracy rose above 70% for globular proteins.

2.2.6.2 PSIPRED

PSIPRED is one of the most successful neural network based contemporary protein secondary structure prediction system [94]. Developed by Jones in 1999, PSIPRED is a two-stage feed forward back propagation neural network with similar structure to PHD. PSIPRED was the first to employ the idea of using the position specific iterated Blast (PSIBLAST) [95] to generate position specific scoring matrix (PSSM) data that was used as direct input to the model instead of single sequence information or simple sequence alignment profile data. PSIPRED was also the first system to use a testing set based on structural similarity criteria rather than sequence similarity criteria. The testing set was non-homologous with the training set, as any protein in the training set with a similar fold to any protein in the testing

set was removed. PSIPRED achieved an average prediction accuracy between 76.5% and 78.3% and continues to be used by many researchers today [96].

There are a number of other successful ANNs that have achieved very good prediction accuracies. For example, SSpro1 and SSpro2 by Pollastri et al, bidirectional recurrent neural networks (BRNNs) to capture long range interactions between residues and overcome some of the limitations associated with small fixed-length input windows of feed-forward networks [97]. A number of consensus methods were developed that were based on studies which proved that the overall prediction accuracies in secondary structure prediction can be improved by combining several estimators [93][98]. For example, Chandonia and Karplus used a combination of up to eight neural networks to increase prediction accuracy up to 76.6% [99]. More recently, Petersen et al have used a jury of up to 800 neural networks to achieve very high prediction accuracy of approximately 77.2%–80.2% [100].

2.2.7 Emerging neural network architectures for secondary structure prediction

2.2.7.1 Segmented memory recurrent neural network (SMRNN)

Two researchers, Chen and Chaudhari have suggested that the performance of RNNs on long-term dependency problem can be improved by using segmented-memory. Accordingly, they have proposed a new neural network architecture that is based on the idea of human memorization of long sequences [101], where people usually divide long sequences into a number of smaller segments. First, they start by memorizing the first segment, then the second segment, after finishing the second segment, they start again from the first segment and cascade it to the second which is newly memorized to assure that they remember both segments together, and this step is repeated till the end of the sequence. So, at the end of each segment, people usually start from the beginning to make sure that they have remembered all previous segments [102].

Chen and Chaudhari called their new model a "segmented memory recurrent neural network" (SMRNN). It is composed of an input layer, an output layer, two hidden layers, and two context layers as illustrated in Figure 2-18. The first hidden layer processes the contextual data associated to symbols, and the second hidden layer processes the contextual data associated to segments. The first and second context layers store the states of first and second hidden layers at preceding cycles respectively [101].

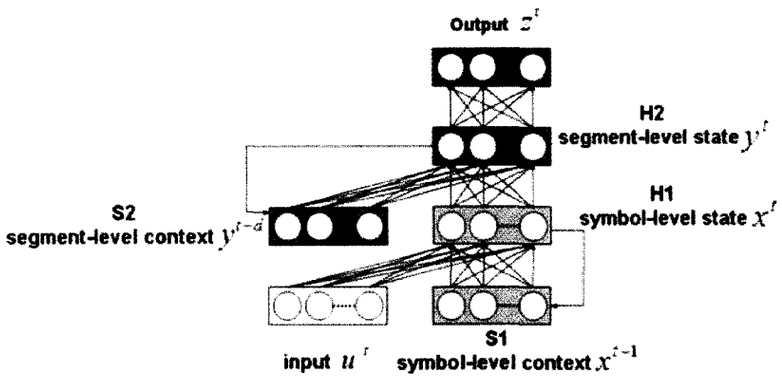

Figure 2-18: Schematic representation of SMRNN (reproduced from reference[101]).

Training of a SMRNN starts by breaking the input sequence into a number of equal segments. Then all symbols in the first segment are applied sequentially to the symbol-level context till the end of this segment. After applying all symbols in the first segment, the symbol-level context is fed to the segment-level context. This procedure is repeated to all the segments till the end of the input sequence. After that, the output of the segment-level context is fed to the output layer in order to obtain the final output. The state of the symbol-level context layer is updated whenever it receives a new symbol from the input sequence, while the state of the segment-level context layer is updated only after it receives a whole segment and also at the end of the sequence. The segment-level state layer cascades all segments sequentially to generate the final sequence [102].

In order to use the SMRNN to predict the protein secondary structure given only the protein primary sequence, the amino acid sequence is presented to the network and is divided into a number of equal segments. Each segment is composed of equal number of amino acids (symbols). All amino acid residues in each segment are presented sequentially to the first hidden layer to update the state of the symbol-level. At the end of each segment, the state of the symbol level is fed to the second hidden layer to update the state of the segment-level. This procedure is repeated to all the segments of the input amino acid sequence. After that, the output of the segment-level context is fed to the output layer in order to obtain the final output.

The performance of SMRNN was assessed and compared to the performance of Elman networks in predicting protein secondary structure for sequences of length 60-200 amino acids. It was found that the SMRNN achieved higher prediction accuracies and was capable of capturing longer ranges of dependencies than Elman networks, as Elman networks had difficulty in learning to classify protein sequences of length 65 or more [101].

2.2.7.2 Bidirectional segmented memory recurrent neural network (BSMRNN)

In order to exploit more information contained in distant portions of protein sequences and in order to capture long range interactions, Chen and Chaudhari have further developed the SMRNN into a new model called the bidirectional segmented memory recurrent neural network (BSMRNN) [101][102][103]. The BSMRNN has overcome the limitation of recurrent neural networks; in which the output of recurrent neural networks depends solely on the values of the previous and current inputs and not on the values of any future inputs. The BSMRNN is composed of three sub networks. The first sub network is a forward segmented-memory recurrent neural network, the second is a backward segmented-memory recurrent neural network, and the third is a multi layer Perceptron. The forward SMRNN and the backward SMRNN are employed to capture both the forward (upstream) and the backward

(downstream) context respectively. Then the multi layer Perceptron is employed to combine both upstream and downstream information. The BSMRNN has greatly improved the performance on protein secondary structure which is a long-term dependency problem and achieved very high prediction accuracy that surpasses 90% [102].

2.3 Previous studies of species-specific characteristics

There are several studies that have investigated the variation of species-specific characteristics across different species. For example, in a study on 64 different species in reference [104]; researchers found that there exists species-specific characteristics that constrain distribution and abundance of these species in different habitats. Also scientists have studied the species-specific predictions of the effect of three characteristics of terrestrial species: mobility, population density and habitat specificity [105]. In references [106][107], It was found that glycoprotein E2 of the hepatitis C virus binds to CD81 on human and chimpanzee cells, but not to mouse, so they suggested that CD81 molecules are species-specific and the presence of these molecules indicate high susceptibility to infection with hepatitis C. And In reference [108], researchers examined a number of species whose genomes have been completely sequenced, and observed that there exists species-specific variation in amino acid composition, and also observed that the variation of amino acids in different species is influenced by several environmental influences (e.g. pH, pressure, salt, and solute concentrations).

2.4 Conclusions

This chapter provides a quick review of protein structure, its hierarchical levels, and describes how three dimensional protein structure can be determined through complex experimental procedures such as X-ray crystallography and nuclear magnetic resonance, and through computational three-dimensional structure prediction methods. Various computational methods have been developed to predict protein secondary structure,

however only neural networks will be used in this thesis. Some of the fundamental concepts of major classes of neural networks were presented in this chapter, with a brief description of their different topologies and learning algorithms. Then the applications of using neural networks in the field of biology were presented with a quick overview of the earliest attempts to build neural networks, particularly to predict protein secondary structure. Then a number of the most successful prediction methods in this field was discussed, including a number of neural-network based methods that achieved prediction accuracies between 76% and 80% [6, 99, 100]. The chapter concluded with an overview of the most recent models that achieved very high prediction accuracies above 80%, including the SMRNN and BSMRNN techniques.

3 Evidence for species-specific protein folding

3.1 Introduction

In this chapter, we will test hypothesis that there exists species-specific aspects to protein folding. The amino acid composition in different species has been previously investigated in reference [108], where it was found that most protein sequence composition variation was explained by GC content (i.e. the proportion of guanine and cytosine bases in the genome vs. adenine and thymine) in the genome and also by environmental niche. To test hypothesis about the presence of species-specific aspects to protein folding, the sequence composition, protein secondary structure composition, and variation in protein chain lengths will be analyzed across 12 different species that have been randomly chosen to represent the six Kingdoms of life. Construction of these species-specific protein datasets will be discussed first, since they are used in the remainder of the thesis.

3.2 The six Kingdoms of life

Over the years, scientists have tried to group living organisms into categories based on similar physical characteristics [109][110][111]. In 1969, a scientist called Robert Whittaker classified all living organisms into a five Kingdom system (Animals, Plants, Fungi, Protists, and Monera) [112]. The five Kingdom system was still based on classifying organisms into groups based on common outward physical and behaviour characteristics. As researchers learnt more about phylogeny and genetic sequencing, the classification of organisms followed a new concept that is based on grouping organisms according to similarities in their ribosomal RNA sequence [113]. According to the new concept, organisms are classified into a three domain system (Eukaryotes, Archaecteria and Eubacteria). The three Domain system has divided the Monera Kingdom into two Domains: Archaecteria and Eubacteria, and also has grouped the four Kingdoms (Animals, Plants, Protista, and Fungi) together into

the Eukaryotes Domain [113], as it was found that these four Kingdoms share similar genetic composition based on RNA studies. Also scientists found that Eukaryotes are genetically more closely related (i.e. have shorter phylogenetic distance) to the Archaebacteria than they are to the Eubacteria [114]. Recently, the three Domain system and the five Kingdom system have been combined into a six Kingdom system (Animals, Plants, Fungi, Protists, Archaecteria and Eubacteria) [115] as illustrated in Figure 3-1.

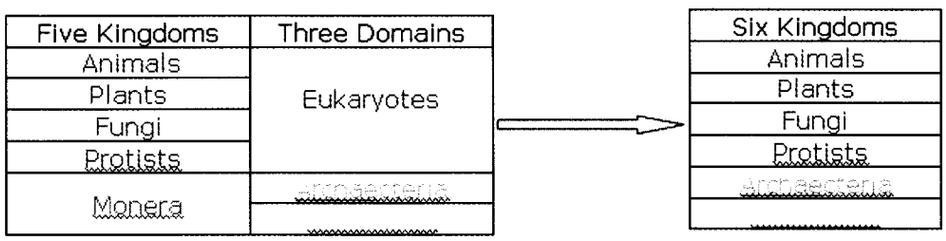

Figure 3-1: Six Kingdoms of life.

3.3 Construction of species-specific datasets

We have created 12 different datasets to represent the six Kingdoms of life (Animals, Plants, Archaebacteria, Eubacteria, Fungi, Protists). In this thesis, Human and mouse represent the Animal Kingdom, *Arabidopsis Thaliana* and *Zea Mays* represent the Plant Kingdom, *Archaeoglobus Fulgidus* and *Pyrococcus Furiosus* represent the Archaebacteria Kingdom, *Bacillus Subtilis* and *Thermotoga Maritima* represent the Eubacteria Kingdom, *Schizosaccharomyces Pombe* and *Saccharomyces Cerevisae* represent the Fungi Kingdom, and finally *Plasmodium Falciparum* and *Trypanosoma Cruzi* represent the Protist Kingdom. The specific species selected to represent each Kingdom were selected randomly, with a bias towards species that have more experimentally solved protein structures available in order to have the largest possible datasets.

For each species, the primary sequence information from the PDB database [116] was parsed on the 17th of April 2009 to extract all sequence for experimentally solved structures for a given species. Sequences that were too short to be analyzed by the neural network were excluded from this study. Then PDB's SOAP web service [117] was used to fetch the actual secondary structure for each species, which is derived from the experimentally solved 3D tertiary structure of the protein as discussed below in section 4.2.1.

3.4 Species-specific sequence composition

The frequency of each amino acid in each species was calculated following the equation in reference [108] which is calculated as :

$$\text{Frequency of amino acid AA in species S} = \frac{\text{Number of counts of amino acid AA in species S}}{\text{Total number of counts for all amino acids in species S}} \times 100$$

Results of this analysis are illustrated in

Table 3-1. The results show a considerable variation of the frequency of each amino acid across the 12 species. For example, the percent of Alanine (A) varies from a minimum of 4.84% in *Plasmodium Falciparum* to a maximum of 8.42% in *Trypanosoma Cruzi*. Figure 3-2 shows a more detailed investigation of the minimum and maximum frequency of the 20 AAs in the 12 species. The results in

Table 3-1 and Figure 3-2 indicate that there exists a significant variation in the species-specific sequence composition across the 12 species.

Species	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
Human	6.87	1.96	5.38	7.08	4.18	6.68	2.96	5.1	6.09	9.84	2.4	3.95	4.9	4.47	5.09	6.69	5.07	6.72	1.31	3.26
Mouse	6.52	1.89	5.16	6.14	4.11	7.2	2.45	4.65	5.75	8.84	2.11	4.03	5.17	4.26	4.73	8.47	6.37	6.76	1.69	3.72
A. Thaliana	7.62	1.41	5.74	6.81	4.4	7.75	2.47	5.64	6.41	8.77	2.3	3.91	4.89	3.03	4.6	7.3	5.33	7.6	1.2	2.83
Z. Mays	8.07	1.3	6.11	6.27	3.75	8.19	2.19	5.33	5.86	8.06	2.31	4.29	5.31	3.41	5.38	5.62	4.79	7.27	1.82	4.66
A.Fulgidus	8.31	1.01	5.31	9.59	4.17	7.45	2.05	7.28	7.05	8.53	2.58	2.91	3.93	1.91	6.14	5.2	3.75	8.61	0.9	3.31
P.Furiosus	8.23	0.36	4.9	10.8	4.21	7.05	1.97	7.36	8.74	9.67	2.41	3.05	4.41	1.93	4.77	4.46	3.45	6.77	1.18	4.3
B. Subtilis	8.13	0.77	5.75	7.97	3.71	7.5	3.06	6.64	6.76	8.63	2.64	4.1	3.75	3.79	4.09	5.78	5.52	6.9	1.02	3.48
T. Maritima	6.15	0.59	5.53	9.24	4.66	7.22	2.81	6.96	7.5	8.94	2.45	3.89	4.01	1.7	5.62	5.18	4.24	8.71	1.07	3.49
S. Pombe	6.75	1.08	5.33	6.89	3.93	6.57	2.65	6.69	6.1	10.2	2.42	4.62	4.88	3.34	4.7	7.24	4.79	6.98	1.22	3.59
S.Cerevisiae	6.86	1.04	6.3	6.99	4.17	6.6	2.28	6.49	7.24	9.25	2.01	5.05	4.26	3.57	4.32	6.88	5.56	6.5	1.09	3.55
P. Falciparum	4.84	1.77	6.09	6.33	4.68	6.24	2.62	7.9	8.98	8.76	2.07	7.71	3.21	2.9	2.9	6.25	4.87	6.39	0.88	4.61
T. Cruzi	8.42	1.68	5.02	6.38	3.91	8.23	2.51	4.59	5.47	8.35	2.53	4.03	4.41	2.83	5.27	6.92	5.69	8.77	1.48	3.51

Table 3-1: Sequence composition of different species.

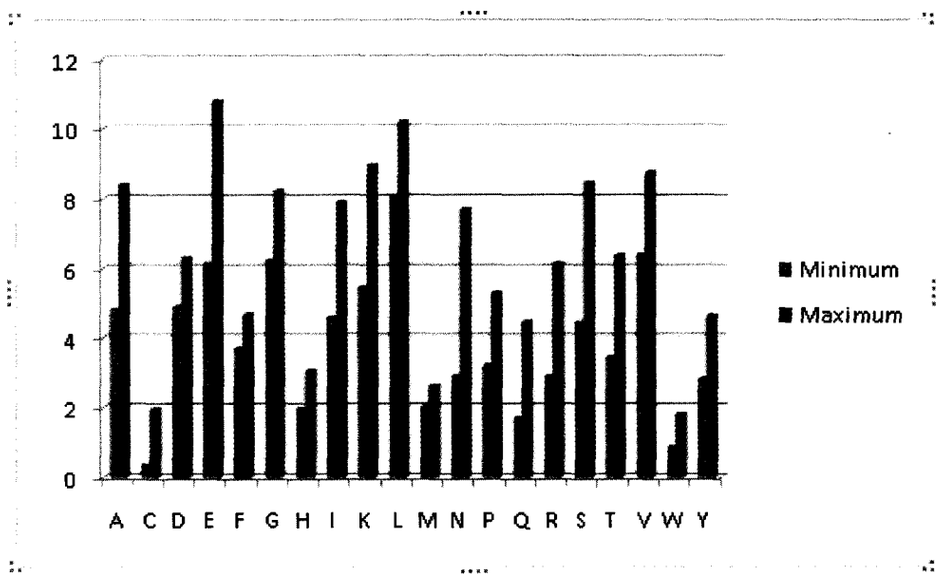

Figure 3-2: The minimum and maximum frequency of the 20AAs in the 12 species.

3.5 Species-specific secondary structure composition

In order to investigate if there exists a variation in the protein secondary structure composition across different species, the frequency of each secondary structure element (helices, beta-strands, and coils) was calculated in each species as follows :

Frequency of secondary structure element EE in species S

$$= \frac{\text{Number of counts of secondary structure element EE in species } S}{\text{Total number of counts for all secondary structure elements in species S}} \times 100$$

Results are illustrated in Table 3-2. The results indicate the presence of variation in the percent of the secondary structure elements across the 12 species. The percent of helices vary from a minimum of 28.39% to a maximum of 46.32% across the 12 species, the percent of beta-strands vary from a minimum of 17.67% to a maximum of 27.83%, and the percent of coils vary from a minimum of 36.01% to a maximum of 43.78%. The results indicate that there exists a species-specific secondary structure difference across the 12 species. Figure 3-3 shows the minimum and maximum frequency of the three secondary

structure elements across the 12 different species. As illustrated in the figure, there exists a significant variation in the secondary structure composition across the 12 species.

Species	Percent of Helices	Percent of Beta-strands	Percent of coils
Human	35.48	21.15	43.37
Mouse	28.39	27.83	43.78
A. Thaliana	37.31	19.85	42.84
Z. Mays	37.41	19.92	42.67
A. Fulgidus	39.25	22.92	37.83
P. Furiosus	46.32	17.67	36.01
B. Subtilis	37.99	22.32	39.69
T. Maritima	41.64	20.13	38.24
S. Pombe	36.56	20.55	42.89
S. Cerevisiae	38.02	18.93	43.05
P. Falciparum	36.81	22	41.19
T. Cruzi	30.7	27.02	42.28

Table 3-2: Variation of secondary structure composition of different species.

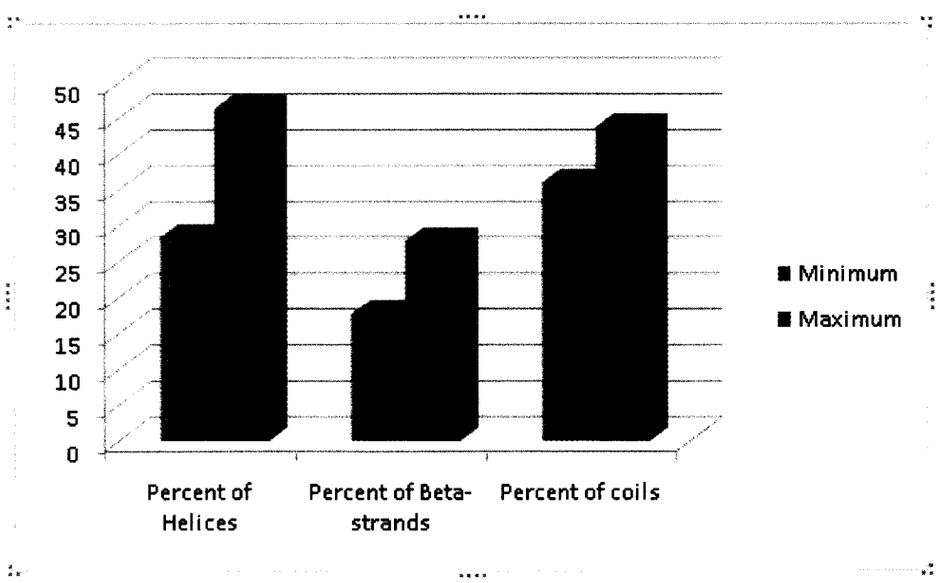

Figure 3-3: Minimum and maximum percent of secondary structure elements across the 12 different species.

3.6 Species-specific average chain lengths

In order to investigate whether chain lengths vary with different species, we created a Perl script to calculate the average chain lengths for the 12 species. Results are shown in Table 3-3. It is apparent that the chain lengths vary widely for different species. For the 12 selected species, the chain lengths vary from a minimum of 234 amino acids in *Bacillus Subtilis* to a maximum of 352 amino acids in *Zea Mays*. The results indicates that there exists a species-specific chain lengths variation across the 12 species. Figure 3-4 illustrates the variation of the average chain lengths for different species.

Species	Average chain length
Human	244
Mouse	236
A. Thaliana	282
Z. Mays	352
A. Fulgidus	245
P. Furiosus	252
B. Subtilis	234
T. Maritima	275
S. Pombe	259
S. Cerevisiae	286
P. Falciparum	288
T. Cruzi	329

Table 3-3: Average chain lengths across the 12 different species.

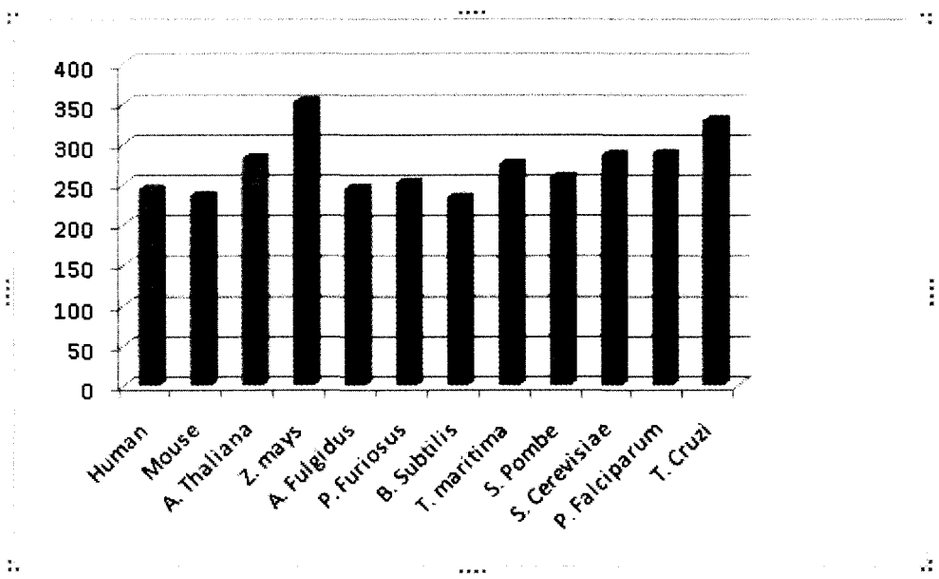

Figure 3-4: Average chain length variation across the 12 different species.

3.7 Conclusions

In this chapter, we have tested the hypothesis that there exists species-specific aspects to protein folding. We have randomly selected 12 species to represent the six Kingdoms of life to test this hypothesis. The results in this chapter indicate that there exists strong evidence for species-specific protein folding, as there is a significant difference in the sequence composition, secondary structure composition, and protein chain lengths across the 12 species.

4 Examination of optimal network architecture for each species

4.1 Introduction

In this chapter we will test hypothesis that, due to the presence of species-specific aspects to protein folding, we expect there would be a particular neural network architecture that yields optimum accuracy for each species. To test this hypothesis, 5 different neural network architectures were trained on 12 different species to investigate the presence of an optimum classifier for each species. Furthermore, we seek to investigate if there exists a variation in the optimum number of hidden nodes for neural networks for each species that yield the highest prediction accuracy.

4.2 Methodology

4.2.1 Encoding scheme

Training and testing datasets are required in training neural networks, where both the input primary amino acid sequence and also the output secondary structure are known *a priori*. Known secondary structure may be computed from experimentally-derived 3D tertiary protein structures from the Protein Data Bank. The Dictionary of Protein Secondary Structure (DSSP) [118] is the most common used method for computing secondary structure data from three dimensional tertiary structure. The DSSP program classifies residues by their hydrogen bonding patterns into eight classes: H (α helix), B (isolated β-bridge), E (extended β-strand), G (3/10-helix), I (π-helix), T (hydrogen bonded turn), S (bend) and C (other). Most of the prediction methods are trained and assessed for only 3 classes associated with helices (H), beta-strands (E), and coils (C). Accordingly in this thesis, I have reduced the 8 classes into the three standard classes: helices, strands, and

coils using a widely used reduction method that was adopted in many references (e.g. [119][120][121]). In this reduction method, H contains the DSSP classes H, G, and I, E contains the DSSP classes E and B, and C contains everything else. Therefore, all neural networks used in this study have three output nodes to represent H (helix), E (extended B-strand), or T (coil).

4.2.2 Generation of evolutionary profiles

Amino acid sequences are represented as strings of letters which cannot be used directly to train neural networks since input nodes require numeric inputs. The input data are therefore extracted from a position specific scoring matrix (PSSM). An outline of the encoding method of the inputs to the neural networks is illustrated in Figure 4-1.

PSSM data are created by giving scores for each of the 20 possible amino acids at each position in the protein sequence; high scores are given to highly conserved positions and very low or negative scores are given to less conserved positions [122]. This converts a string of length *n* into a numeric matrix with *n* rows and 20 columns (i.e. one column for each of the 20 possible amino acids at each position). These numeric values can be calculated from the PSSM generated by PSI-BLAST. PSI-BLAST is a very powerful sequence searching method that creates evolutionary profiles [122]. Many protein secondary structure methods have used PSI-BLAST due to its ability to identify distant homologs (i.e. distantly related proteins) even when sequence similarity is low [123][124]. Furthermore, it was found that the utilization of divergent evolutionary profiles created by PSI-BLAST leads to a consistent improvement in the prediction accuracy [125][126]. In order for the PSI-BLAST to create very sensitive profiles, it necessitates a pre-filtered database to search against with no repeated sequences [123]. So in this thesis, I have downloaded the NCBI non-redundant protein sequence database [127] and it was filtered using the pfilt program [123]. This latter filtering step removes any non-globular (e.g. coiled-coil) regions since

they differ significantly from the globular protein domains that are the focus of this study. Also, low complexity regions (e.g. simple repeats of a small number of amino acids) are removed from the sequence database to prevent meaningless sequence matches.

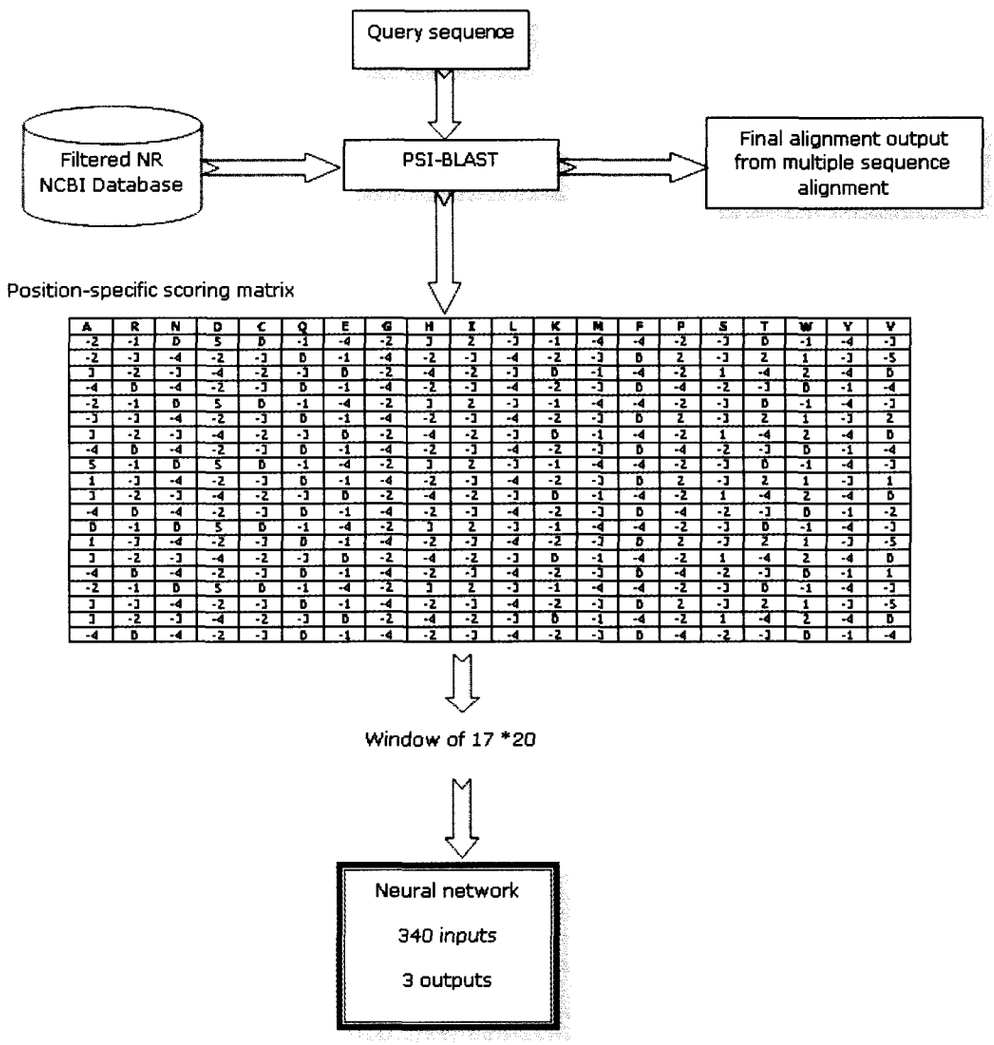

Figure 4-1: An outline of creating the neural network inputs using PSSM created by PSI-BLAST.

For all neural networks implemented in this study, protein sequence windows with a width of 17 amino acids were used as this has been found by many previous studies to be an optimal window width [128][129][130][131]. In order to use the output data of the PSSM,

each amino acid was represented by 20 numbers which gives a total of 17*20=340 input nodes.

4.2.3 Removal of homologous proteins in training and testing datasets

In order to prevent overestimated prediction accuracy, there must be no homologous proteins in the training and testing datasets. So the PISCES server [132] was used to ensure that there are no proteins in the training dataset with significant degree of sequence similarity to proteins in the testing dataset. PISCES is a protein sequence culling server that culls sets of protein sequences from the Protein Data Bank by sequence identity and structural quality criteria [132]. In this thesis, PISCES was used to remove protein sequences that share more than 25% sequence identity. Furthermore, only PDB structure obtained by X-ray diffraction methods with maximum resolution of 3 A° were used in this study. Table 4-1 shows the 12 species that were randomly chosen from the six Kingdoms of life, the number of protein chains before and after filtering using the PISCES, and the number of amino acid residues for each species after using PISCES.

Kingdom	Species	Protein chains before PISCES	Protein chains after PISCES	Amino acid residues
Animals	Human	20706	1156	262595
	Mouse	3129	317	69352
Plants	*A. Thaliana*	655	112	29724
	Z. Mays	159	22	7372
Archaebacteria	*A. Fulgidus*	405	89	20265
	P. Furiosus	477	80	18795
Eubacteria	*B.Subtilis*	1501	264	57200
	T. Maritima	1030	201	51850
Fungi	*S. Pombe*	202	38	108003
	S.Cerevisae	3187	401	9192
Protists	*P. Falciparum*	323	50	13277
	T.Cruzi	179	26	8126

Table 4-1: Number of protein chains and amino acid residues in 12 species.

4.2.4 Evaluation of prediction accuracy

In this thesis, the Q3 score was used to evaluate the performance of the neural networks. Q3 is one of the most common evaluation methods of secondary structure prediction. It calculates the percentage of correctly predicted residues as coil, helix, or strand. The equation used to compute Q3 is defined as:

$$Q_3 = \frac{Total\ number\ of\ correctly\ predicted\ residues}{Total\ number\ of\ residues} \times 100\ \% \qquad \text{equation (4-1)}$$

4.3 Results

4.3.1 Optimum neural network architecture

We have trained 5 different neural network architectures (FFNN ,Elman, NARX, RNN, and PNN) that were discussed in section 2.2.4 on 12 different species-specific datasets. As mentioned above, the MATLAB neural network tool box [152] was used to create all neural networks used in this thesis. The neural network toolbox randomly divides the input data into three different data sets. The first data set comprises 60% of the input data and it is used to train the network. The second data set comprises 20% of the input data and it is used to validate how well the network has generalized during training. The training procedure continues as long the training decreases the network's total error on the validation data. As soon as the network memorizes the training set, training is halted, thus avoiding the problem of over fitting [156]. The last data set comprises 20% of the input data and it is used to provide an independent test of network generalization.

The 5 neural network architectures were tested using a 5-fold cross validation protocol test, where 1/5th of the data was used for testing and the remaining 4/5th of the data was used for training. This was repeated 5 times such that each network was trained and tested on different subsets of the data. The results are shown in Table 4-2, where the best performing network is highlighted in bold for each species.

Species	Elman	FFNN	NARX	RNN	PNN
Human	70.3	**71.67**	71.61	71.15	37.89
Mouse	70.35	**70.91**	70.88	70.49	38.18
A. Thaliana	**72.55**	72.33	72.21	71.54	37.95
Z. Mays	**70.82**	69.1	69.5	66.07	39.77
A. Fulgidus	**69.57**	68.93	69.08	68.14	40.11
P. Furiosus	**70.74**	69.99	70.24	69.25	40.06
B. Subtilis	**74.54**	74.16	74.13	73.71	38.15
T. Maritima	**74.1**	73.65	73.58	72.92	40.9
S. Pombe	**70.57**	69	68.89	63.71	35.64
S. Cerevisiae	71.21	**72.53**	72.46	71.93	40.5
P. Falciparum	**71.65**	71.24	71.06	68.61	39.6
T. Cruzi	**72.85**	72.37	72.23	68.76	37.75

Table 4-2: Results of 5 different neural network architectures trained on 12 species.

The results show that all network architectures perform relatively well, except for PNNs. In particular, historical networks represented by Elman networks appear to be the best for the majority of species and perform relatively well on the remainder. The high prediction accuracy achieved by Elman networks may be due to the presence of the context layer that is capable of capturing previous states of the sequential context information stored in the hidden layers. This has a direct impact on the protein secondary structure prediction as protein sequences and conformations are strongly affected by the previous secondary structure states [160]. Thus Elman networks are an effective neural network architecture that is capable of capturing sequential secondary structure correlations between adjacent residues. On the other hand, results in Table 4-2 reveal that PNN have performed very poorly and have achieved the lowest prediction accuracies for all the species. This may be due to the fact that the learning algorithm in PNNs is not iterative and the weights are not randomized (the learning occurs in only one iteration that produces fixed weights) [161]. Due to the relatively high performance of Elman networks on all species, I will use them for the remainder of all experiments performed in this thesis.

4.3.2 Optimum number of hidden nodes for Elman network

It was shown in section 4.3.1 that Elman network has achieved the highest prediction accuracies for the majority of species. Accordingly for this section and the following chapters of the thesis, we will focus only on investigating Elman networks. The number of hidden nodes for Elman networks across the 12 species will now be investigated to assess whether the optimum number of hidden nodes are species-specific. In the next chapter, species-specific Elman networks will be trained for each species and they will be compared with Elman networks trained using a pool of proteins from multiple species to assess whether training a neural network with species-specific would yield higher prediction accuracies than when trained on a pool of mixed species.

A five-fold cross validation testing was performed to assess whether there exists a variation in the optimum number of hidden nodes across the 12 species. For each species, the number of hidden nodes of the Elman networks was increased from 3 to 50 nodes. The results are shown in Table 4-3. The results show that there exists an optimum number of hidden nodes for Elman network for each species that yield the highest prediction accuracy.

Also the results in Table 4-3 illustrate that for each species, the prediction accuracy initially increases as the number of hidden nodes are incremented, as a small number of hidden nodes might not be enough to permit adequate interpolation between samples [79][123]. Then the prediction accuracy starts to gradually decrease with further increasing of the number of hidden nodes, as too many hidden nodes may cause the network to memorize the training patterns instead of achieving a satisfactory level of generalization and becomes "over trained" [79][123]. For the remaining experiments, the number of hidden nodes that provided the highest accuracy was used for each species. A better choice may have been to

use only 3 hidden nodes for all species, since an optimal classifier is the least complex system that achieves near-optimal accuracies [156].

Species	3	5	10	20	30	40	50
Human	69.00	70.20	**70.40**	50.89	44.16	50.40	49.61
Mouse	70.21	70.37	70.63	**70.73**	57.38	56.21	52.07
A. Thaliana	72.29	72.11	72.25	**72.55**	59.15	58.89	45.43
Z. Mays	70.31	69.94	**70.82**	70.22	70.10	48.64	38.51
A. Fulgidus	69.11	69.36	69.45	**69.57**	69.10	56.11	50.22
P. Furiosus	70.22	70.26	**70.74**	71.30	64.1	64.68	42.23
B. Subtilis	74.14	74.4	74.67	**74.7**	66.43	59.52	43.00
T. Maritima	73.69	73.96	74.11	**74.17**	73.93	43.19	49.35
S.Pombe	69.6	69.46	69.42	**70.57**	63.23	63.61	55.71
S. Cerevisae	71.22	71.26	71.39	**71.47**	43.63	58.42	49.23
P. Falciparum	71.20	70.79	71.44	71.59	**71.65**	55.5	41.92
T. Cruzi	72.00	**72.85**	72.62	72.52	58.00	48.00	56.00
Mixed	71.56	71.67	71.89	**72.18**	36.86	51.73	49.83

Table 4-3: Accuracy vs. number of hidden nodes.

Table 4-4 summarizes the results of Table 4-3 and shows the optimum number of hidden nodes and the associated accuracy for each species. The results of Table 4-3 and Table 4-4 show that there is no correlation between optimal number of hidden nodes and phylogentic relations. For example, the 2 Protist species (*P. Falciparum* and *T. Cruzi*) have widely different number of optimal hidden nodes. Furthermore, the results of training Elman network with species-specific has surpassed the results in reference [162], where an Elman network was trained on a pool of multiple species.

Kingdom	Species	Optimum # nodes	Accuracy
Animals	Human	10	70.4
	Mouse	20	70.41
Plants	*A. Thaliana*	20	72.55
	Z.Mays	10	70.82
Archaebacteria	*A. Fulgidus*	20	69.57
	P. Furiosus	10	70.74
Eubacteria	*B. Subtilis*	20	74.7
	T. Maritima	20	74.17
Fungi	*S. Cerevisae*	20	71.47
	S. Pombe	20	70.57
Protists	*P. Falciparum*	30	71.65
	T. Cruzi	5	72.85

Table 4-4: Summary of optimum number of hidden nodes for each species.

4.4 Conclusions

In this chapter we have explored whether there is a particular neural network architecture that yields optimum accuracy for each species. The results indicate that a number of neural network architectures perform well on all species, except for PNNs. Elman networks was found to be the optimal or near-optimal architecture for all species. Also the results in this chapter reveals that there exists a variation in the optimum number of hidden nodes for Elman networks for each species that yield the highest prediction accuracy.

5 Species-specific protein secondary structure prediction

5.1 Introduction

Seeing that there are species-specific differences in protein sequence and structure, we hypothesise that protein secondary structure prediction methods trained on species-specific datasets will outperform classifiers trained on mixed species datasets when the training and testing proteins come from the same species. As seen in Chapter 4, the Elman network architecture has achieved the highest prediction accuracies for most of the species, and performed satisfactorily on the remainder. Accordingly in this chapter, we will focus only on investigating Elman networks. Elman networks will be trained on one species at a time. Structure prediction accuracy will be compared to a generic classifier built using data from a pool of multiple species to assess whether training a neural network with species-specific data will yield higher prediction accuracies than when trained on a pool of mixed species. Also in this chapter, we will test the hypothesis that when an exact match between training and testing species is not possible, predictors trained on phylogenetically related species will outperform other species-specific and generic predictors. Lastly, voting among a pool of species-specific classifiers will be demonstrated to achieve the highest prediction accuracy. We begin this chapter by briefly describing an experimental protocol that did not support our hypothesis before moving on to the final experiment design.

5.2 Evaluation of species-specific predictors using orthologous proteins

In order to investigate the hypothesis that species-specific classifiers will provide the improved prediction accuracy, we initially ran extensive experiments using datasets of orthologous proteins. A pair of orthologous proteins describes two proteins from two

different species that are both evolved from the same ancestral protein and that continue to serve the same function [166]. An example of this would be haemoglobin in humans and cows, where the protein sequence, structure, and function are conserved in both species. Motivation for using othologous pairs of proteins was to control for sequence length and composition in species-specific protein datasets. A tool called InParanoid [165] was downloaded and applied to each pair of proteins to extract the subsets of proteins which formed orthologous pairs. Elman networks were trained and tested for every combination of species. These analyses took extensive effort requiring multiple customized PERL scripts and hundreds of hours of CPU computation time. Unfortunately, this experimental approach failed to provide evidence to support our hypothesis. Results showed no systematic difference in classification accuracy when proteins from one species were tested using species-specific predictors from the same and different species. Details of the methodology and results, as well as a discussion of why this experimental protocol ultimately failed to support our hypothesis, are given in Appendix B. Although this was effectively a 'dead end' in terms of proving our hypothesis, these experiments represent significant effort, and these results may be useful to future researchers examining this area of protein structure prediction. Hence, we have retained the materials describing these experiments in an appendix.

An analysis of the subset of proteins used for these experiments (i.e. only those 5-10% of all proteins which form orthologous pairs) showed that they were atypical in terms of sequence and structure composition. Both sequence and structure were highly conserved. We believe this is due to the fact that these orthologous proteins are so highly conserved in function that the species-specific elements of protein folding are overshadowed by the constraints imposed by the protein function. In the remainder of this chapter, all experiments will be performed using more general protein datasets, no longer restricting our analysis to using only orthologous proteins across different species.

5.3 Comparison of species-specific vs. generic predictors

5.3.1 Creation of species-specific and mixed species datasets

A new mixed dataset was created which is composed of randomly chosen proteins from the 12 species used throughout this thesis. The number of protein chains for each species and also the number of protein chains from each Kingdom are shown in Table 5-1. The mixed dataset is composed of 456 protein chains (76 protein chains from each species). After that the PISCES tool [132] was applied to the entire mixed data set to remove any identical protein chains in the dataset, resulting in a final a mixed dataset composed of 396 protein chains.

Kingdom	Species	Protein chains	Protein chains in mixed species	Protein chains from each Kingdom
Animals	Human	1156	38	76
	Mouse	317	38	
Plants	*A. Thaliana*	112	54	76
	Z.Mays	22	22	
Archaebacteria	*A. Fulgidus*	89	38	76
	P. Furiosus	80	38	
Eubacteria	*B. Subtilis*	264	38	76
	T. Maritima	201	38	
Fungi	*S. Pombe*	38	38	76
	S. Cerevisae	401	38	
Protists	*P. Falciparum*	50	50	76
	T. Cruzi	26	26	

Table 5-1: Number of protein chains from the 12 species used to create a mixed dataset.

We created Perl scripts to examine the species of origin for each protein chain in the mixed dataset and to pull out the proteins from each species. This was used as the species-specific and mixed datasets. For example a Perl script was created to pull out human proteins from the mixed dataset and to create 2 new datasets, one dataset is composed only from human proteins, and the second dataset is a mixed dataset composed of all other proteins without the human proteins. This has resulted in 24 datasets, 12 datasets are composed of proteins

from each of the 12 species, and the other 12 datasets are the corresponding mixed species datasets. The number of protein chains in each dataset is shown in Table 5-2. From Table 5-2, it can be seen that human dataset is composed from 35 proteins and the mixed dataset without the human proteins is composed from 361 proteins. These numbers are lower than would be expected from examination of Table 5-1 since application of the PISCES sequence culling tool has removed some sequences from each species.

Kingdom	Species	Protein chains of species A	Proteins in mixed species without A	Protein chains from each Kingdom
Animals	Human	35	361	70
	Mouse	35	361	
Plants	*A. Thaliana*	50	346	69
	Z.Mays	19	377	
Archaebacteria	*A. Fulgidus*	31	365	64
	P. Furiosus	33	363	
Eubacteria	*B. Subtilis*	36	360	71
	T. Maritima	35	361	
Fungi	*S. Pombe*	36	360	66
	S. Cerevisae	30	366	
Protists	*P. Falciparum*	35	361	56
	T. Cruzi	21	375	

Table 5-2: Number of protein chains of the 24 datasets.

5.3.2 Training Elman networks on species-specific and mixed species

Elman networks were used to investigate whether training a neural network with species-specific data would yield higher prediction accuracy than when training networks with data from a pool of multiple species. Elman network was chosen as it has achieved the highest prediction accuracy as shown in Table 4-2. Furthermore, training Elman networks required less computational time than other network architectures. A five-fold cross validation experiment design was used to estimate the prediction accuracy, where 80% of the data was used for training and the remaining 20% of the data was used for testing, and this was repeated 5 times. In this section, the optimum number of hidden nodes from Table 4-3 was used for each species.

For each species, S, an Elman network, N_S, was first trained on a species-specific dataset, D_S, then a second network, $N_{\bar{S}}$, was trained on the corresponding dataset of mixed species with species S excluded, $D_{\bar{S}}$. Then both networks were tested on an independent test dataset from species S. For example, an Elman network was first trained on 80% of the dataset of human proteins and then another Elman network was trained on a dataset of mixed species without human proteins. Both networks were then tested on the remaining 20% of human proteins. This was repeated five times to implement 5-fold cross-validation. Results are illustrated in Table 5-3, where the accuracy of the 5-fold cross validation is reported (average ± standard deviation).

Species	Train	Test	Accuracy
Human	Human	Human	70.12 ± 2.19
	Mixed	Human	68.6 ± 1.54
Mouse	Mouse	Mouse	68.24 ± 2.14
	Mixed	Mouse	67.39 ± 2.25
A. Thaliana	*A. Thaliana*	*A. Thaliana*	72.42 ± 0.77
	Mixed	*A. Thaliana*	72.00 ± 1.56
Z. Mays	*Z. Mays*	*Z. Mays*	70.71 ± 4.59
	Mixed	*Z. Mays*	68.94 ± 3.51
A. Fulgidus	*A. Fulgidus*	*A. Fulgidus*	69.41 ± 2.23
	Mixed	*A. Fulgidus*	67.00 ± 3.32
P. Furiosus	*P. Furiosus*	*P. Furiosus*	67.33 ± 3.26
	Mixed	*P. Furiosus*	65.75 ± 5.63
B. Subtilis	*B.Subtilis*	*B.Subtilis*	72.58 ± 2.42
	Mixed	*B.Subtilis*	71.92 ± 1.69
T. Maritima	*T. Maritima*	*T. Maritima*	71.37 ± 2.43
	Mixed	*T. Maritima*	70.47 ± 2.84
S. Pombe	*S. Pombe*	*S. Pombe*	70.37 ± 2.47
	Mixed	*S. Pombe*	69.51 ± 2.94
S. Cerevisae	*S. Cerevisae*	*S. Cerevisae*	70.66 ± 1.71
	Mixed	*S. Cerevisae*	68.59 ± 1.85
P. Falciparum	*P. Falciparum*	*P. Falciparum*	71.14 ± 3.11
	Mixed	*P. Falciparum*	69.28 ± 3.21
T. Cruzi	*T. Cruzi*	*T. Cruzi*	72.27 ± 1.53
	Mixed	*T. Cruzi*	70.47 ± 1.38

Table 5-3: Elman networks trained on species-specific and mixed species.

Results in Table 5-3 show that when training an Elman network with species-specific and mixed species data, then tested both networks on same dataset, networks trained with species-specific data yield higher prediction accuracies than those trained with mixed species. We conducted a paired t-test between species-specific accuracies and mixed species accuracies, and found that species-specific accuracies are significantly higher (p-value=0.0001). The results are very promising and offer a new effective approach for the prediction of protein secondary structure.

5.3.3 Discussion

To test the hypothesis that protein secondary structure classifiers trained on species-specific datasets will outperform classifiers trained on mixed species datasets, Elman networks were trained with 12 species-specific datasets and 12 datasets of mixed species. We found in all 12 species that Elman networks trained with species-specific datasets outperformed Elman networks trained with mixed species. The results reveal that the optimum classifier for a certain species would be the classifier trained on same species and that classifier will likely outperform a classifier trained on mixed species. For example, if we are to find the best classifier to predict human proteins, that classifier is probably a classifier trained on human proteins. It should be noted that the mixed species data sets in each case did not include any training proteins from the test species. For example, when testing human proteins, the mixed data set contained only non-human proteins. If human proteins were kept in the mixed training set, one may expect higher testing accuracy for the mixed species classifier, even if no identical human proteins existed in the mixed species training and human test dataset.

5.4 Species-specific predictors applied to new species

This section addresses the question of what would be the best classifier to use when the test protein does not come from one of the 12 pre-defined training species. For example if we

are to find an optimum classifier to predict proteins from rat, it is expected that the optimum classifier would be the one trained on other phylogenetically similar species in the Animal Kingdom or from closely related Kingdoms. To test this hypothesis, we have created 16 new datasets to test the 12 classifiers trained on the 12 species described above, plus one classifier trained on mixed species.

5.4.1 Methods

Preparation of training and testing datasets

To train the Elman networks, we have used 12 datasets representing the 12 species that are used throughout this thesis and one mixed dataset containing random proteins selected from the 12 species. For testing the classifiers, we have created 16 new test datasets from randomly selected species from the 6 Kingdoms. From the Animal Kingdom, Gallus Gallus, Bos Taurus, Rattus Norvegicus, and Drosophila Melanogaster were randomly chosen; from the Plant Kingdom, Canavalia Ensiformis, Glycine Max, and Pisum Sativum were randomly chosen; from the Archaebacteris Kingdom, Halobacterium Salinarum, Methanocaldococcus Jannaschit, Sulfolobus Solfataricus, and Thermoplasma Acidophilum were randomly chosen; from the Eubacteria Kingdom, Aquifex Aeolicus, and Haemophilus Influenzae were randomly chosen; from the Fungi Kingdom, Aspergillus Oryzae, and Candida Albicans were randomly chosen; and from the Protist Kingdom, Dictyostelium Discodeum was randomly chosen.

Results of training and testing Elman networks

To prevent the bias of the number of training patterns, we have used an equal number of 6000 AAs to train the classifiers for each of the 12 species and the mixed species. Also, 85 proteins from each of the 16 new test species were randomly chosen to evaluate the 13 classifiers. So in order to find the optimum classifier for each of the 16 new species, 13 classifiers were trained on the 12 species plus the mixed species. Results of species-specific prediction accuracies for only one species are shown below in Table 5-4. Detailed results of

species-specific prediction accuracies for the 15 remaining species are included in Appendix C. Table 5-4 shows the prediction accuracy for 85 test proteins from a species from the animal Kingdom (*Gallus Gallus* or chicken), when all 12 species-specific protein structure prediction methods, plus the mixed species predictor, are applied to the test data. Here, the highest classification accuracy is achieved by the predictor trained on human proteins (shown in bold) which is a species from the same Kingdom (Animal). This same pattern was observed for most of the 16 test species.

Kingdom	Species	Accuracy
Animals	**Human**	**67.94**
	Mouse	66.42
Plants	*A. Thaliana*	67.47
	Z. MaysMays	66.22
Archaebacteria	*A. Fulgidus*	62.2
	P. Furiosus	66.01
Eubacteria	*B. Subtilis*	66.41
	T. Maritima	67.18
Fungi	*S. Pombe*	67.35
	S. Cerevisiae	67.79
Protists	*P. Falciparum*	67.23
	T. Cruzi	67.31
Mixed_species		62.69

Table 5-4: Testing on Gallus Gallus (Animal Kingdom).

Table 5-5 presents a summary for the best and second-best classifier for each of the 16 new species with the Kingdom of each species clearly stated in square brackets. For 14 of the 16 new test species, the best or second-best predictor was trained on a species from the same kingdom. Note that the generic predictor trained on the mixed species datasets never appears as the best nor second-best performing predictor.

Test Species	1st Optimum Classifier	Accuracy	2nd Optimum Classifier	Accuracy
G. Gallus [Animal]	Human [Animal]	67.94	*S. Cerevisiae* [Fungi]	67.79
B. Taurus [Animal]	Mouse [Animal]	67.67	*T. Cruzi* [Protist]	67.35
R. Norvegicus [Animal]	Mouse [Animal]	68.97	*A. Thaliana* [Plant]	68.68
D. Melanogaster [Animal]	*A. Thaliana* [Plant]	68.22	*S. Pombe* [Fungi]	68.11
C. Ensiformis [Plant]	Human [Animal]	67.15	*T. Cruzi* [Protist]	65.68
G. Max [Plant]	*A. Thaliana* [Plant]	64.32	Mouse [Animal]	64.07
P.Sativum [Plant]	*S. Pombe* [Fungi]	71.06	*Z. Mays* [Plant]	70.79
H. Salinarum [Archaebacteria]	*A. Fulgidus* [Archaebacteria]	70.52	*P. Falciparum* [Protist]	70.50
M. Jannaschit [Archaebacteria]	*A. Fulgidus* [Archaebacteria]	70.39	*P. Furiosus* [Archaebacteria]	69.85
S. Solfataricus [Archaebacteria]	*A. Fulgidus* [Archaebacteria]	72.81	*T. Maritima* [Eubacteria]	72.59
T. Acidophilum [Archaebacteria]	*P. Furiosus* [Archaebacteria]	72.45	*T. Maritima* [Eubacteria]	71.57
A. Aeolicus [Eubacteria]	*B. Subtilis* [Eubacteria]	72.07	*T. Maritima* [Eubacteria]	71.80
H. Influenzae [Eubacteria]	*T. Maritima* [Eubacteria]	70.13	*A. Fulgidus* [Archaebacteria]	70.10
A. Oryzae [Fungi]	*S. Cerevisiae* [Fungi]	73.96	Human [Animal]	73.82
C. Albicans [Fungi]	*S. Pombe* [Fungi]	68.74	*A. Thaliana* [Plant]	68.12
D. Discodeum [Protist]	*T. Cruzi* [Protist]	68.93	*S. Pombe* [Fungi]	68.26

Table 5-5: Summary of species-specific prediction accuracy for 16 new test species

5.4.2 Discussion

To test the hypothesis that classifiers from phylogenetically similar species will perform the best, Elman networks were trained on the 12 species plus one mixed dataset, then 16 datasets of new species were created to test the 13 classifiers. The results in Table 5-5 indicate that there exists a slight advantage of training classifiers with phylogenetically similar species to that of the test species. Reference [115] discussed that the 4 Kingdoms: Animals, Plants, Protista, and Fungi Kingdoms are closely related Kingdoms as they share similar genetic composition, and also discussed that these four Kingdoms are genetically more closely related to the Archaebacteria Kingdom than they are to the Eubacteria Kingdom. The results in Table 5-5 reveal that 13/16 species showed strong preference for same Kingdom, 1/16 had 2nd best from same Kingdom, 2/16 had 'closely related' Kingdoms as best. So the optimum classifier for 81% of the species is a classifier trained on a species from the same Kingdom which support the hypothesis that classifiers from phylogenetically similar species will perform the best.

5.5 Voting among classifiers to improve the accuracy

Several secondary structure prediction methods have benefited from using a jury decision (i.e. voting) among a number of classifiers trained on different datasets [97][98]. For example, Chandonia and Karplus have used a combination of up to eight neural networks resulting in an increase in the prediction accuracy up to 76.6% [99]. Also Petersen et al have used a jury of up to 800 neural networks with voting to achieve very high prediction accuracy of approximately 77.2%–80.2% [100]. While all of these studies used mixed species datasets, I expect that voting among classifiers trained on a number species-specific datasets will improve the prediction accuracy. Therefore, we trained 12 Elman networks on the 12 species described above, then tested these 12 classifiers, plus a 13th mixed species classifier, on the 16 new species. First we used voting on the 2 species from same Kingdom

as test species, plus the mixed species. Then, we used voting on all species from same Domain as the test species. Finally, we used all 13 classifiers to vote. Results are illustrated in Table 5-6. The results indicate that voting among classifiers trained on species-specific datasets will definitely improve the accuracy. For example, when G. Gallus proteins were tested, I used voting on the 2 classifiers trained on Human and Mouse proteins (i.e. species from same Kingdom), the accuracy was 69.86% which is higher than the optimum prediction accuracy obtained when an Elman network was trained on either Human proteins or Mouse proteins alone where they achieved an accuracy of 67.94% and 66.42% respectively as shown in Table 5-5. Also, when we used voting on classifiers trained on species from the same Domain (Eukaryotes), the accuracy has increased to 71.7%. Lastly, voting among all 13 classifiers has achieved the highest accuracy (71.74%). We then conducted a paired t-test between the accuracy of voting among 13 classifiers and accuracy of training Elman networks with mixed species, and found that voting accuracy is significantly higher (p-value=9.4×10^{-7}).

Test Species	Accuracy Same Kingdom	Accuracy Same Domain	Accuracy 13 Classifiers
G. Gallus	69.86	71.70	71.74
B. Taurus	69.28	70.96	71.11
R. Norvegicus	70.37	71.5	71.54
D. Melanogaster	70.68	71.27	71.11
C. Ensiformis	63.31	64.58	66.38
G. Max	65.18	66.01	66.01
P.Sativum	72.62	73.28	73.93
H. Salinarum	73.77	73.77	74
M. Jannaschit	71.17	71.17	71.98
S. Solfataricus	73.91	73.91	75.14
T. Acidophilum	73.23	73.23	75.01
A. Aeolicus	72.71	72.71	74.15
H. Influenzae	71.63	71.63	72.76
A. Oryzae	75.95	75.94	76.1
C. Albicans	69.32	71.29	71.42
D. Discodeum	69.85	72.37	72.37

Table 5-6: Voting among 2 species in test Kingdom, species in test Domain, and voting on all 13 classifiers.

5.6 Conclusions

In this chapter, we have tested the hypothesis that protein secondary structure prediction methods trained on species-specific datasets will outperform classifiers trained on mixed species datasets. Results support the hypothesis and indicate that classifiers trained on species-specific datasets perform better than classifiers trained on a pool of mixed species datasets when an exact match between training and testing species is available. Also, we have tested the hypothesis that training classifiers using species-specific datasets from phylogenetically similar species to the test species will perform the best. The results support this hypothesis, with 14 of 16 test species showing a preference for predictors trained using species from the same Kingdom. Finally, we showed that voting among classifiers trained on species-specific datasets will result in a further improvement in the prediction accuracy.

6 Thesis Summary and Future Recommendations

6.1 Summary of contributions

In this thesis, we built species-specific dataset for 28 different species. This dataset was used for the experiments in this thesis, and can be used in other research studies that is based on species-specific characteristics. We found strong evidence for species-specific aspects to protein synthesis and folding based on the variability of sequence and structure composition and protein chain length between species. Based on the study on five different neural network architectures (feed-forward networks, recurrent neural networks, nonlinear autoregressive networks with exogenous inputs, Elman networks, and probabilistic neural networks), we explored whether there was an optimal neural network architecture for each species. We found that historical networks represented by Elman networks achieve the highest prediction accuracies for most of the species, while radial basis networks, represented by probabilistic neural networks, have achieved the lowest prediction accuracies for all the species. While a single architecture (i.e. Elman networks) appeared to work uniformly among all species, there was a species-specific preference for the number of hidden nodes in each network.

It was demonstrated that predictors trained on the same species as the test species uniformly outperform generic structure predictors trained using pooled data from mixed species. Furthermore, we showed that when predicting the structure of a protein from a given species, there is preference to predictors trained on closely related species. Lastly, we showed that voting among several species-specific classifiers provides the highest classification accuracy.

Taken together, this thesis represents the first reported results that shows that species-specific neural network predictors will be more effective than predictors trained on protein data pooled from multiple species, and there is preference to predictors trained on phylogenetically related species.

6.2 Recommendations for future work

The main goal of protein structure prediction methods is to predict the three-dimensional structure of proteins given knowledge only of its primary sequence. Since the direct prediction of protein tertiary structure is a very challenging problem, thus prediction of protein secondary structure is employed as a starting point to predict protein tertiary structure. Thus the application of species-specific predictors can be employed as an effective starting point to species-specific prediction of protein tertiary structure. The experiments in this study were performed solely on a simple Elman network, so training more effective predictors that can achieve accuracies above 80% (for example Bidirectional recurrent neural networks [97], Segmented memory recurrent neural networks [101], and Bidirectional segmented memory recurrent neural networks [102][103]) is a worthy goal for future studies. Voting to combine several species-specific predictors using these more advanced network architectures are also expected to perform very well and should be investigated.

In this thesis, it was demonstrated that there exists species-specific aspects to protein folding based on the variability in sequence and structure composition and variation in average chain length across different species. These species-specific aspects could be further investigated by analyzing different window widths for each species. It is expected that some species may require longer windows than others.

In this thesis, we showed that when predicting the structure of a protein from a given species there is preference to predictors trained on phylogenetically related species. These findings could be extended to quantify the phylogenetic distance and correlating it with accuracy.

Lastly, a web service can be implemented for species-specific protein secondary structure prediction.

Appendix A: Detailed description of the backpropagation algorithm

Appendix A includes a detailed mathematical description of the backpropagation algorithm (following references [79][80][145]) used to train feedforward neural networks. The goal of neural network training algorithms is to optimize the interconnection weights between nodes in the network such that training inputs generate outputs as close as possible (typically in the mean-squared sense) to the training output data. Note that other network architectures, such as the Elman networks used extensively in this thesis, also use variants of this algorithm to optimize their interconnection weights.

A.1 Backpropagation learning algorithm

The backpropagation term illustrates how the network is trained. The backpropagation algorithm was first proposed in 1974 by Werbos [80]. Back propagation is a supervised training algorithm, where the network is provided with inputs and their associated outputs. These associated outputs will be compared against the actual output from the network. The error between the actual and the desired output will be used by the back propagation learning algorithm to adjust the connection weights in a backward direction.

Training a feed-forward network using the back propagation learning algorithm starts by presenting patterns to the network starting at the input layer, then propagating the signal forward through the hidden layers, after that the output is calculated. After that a feedback signal which represents the error between the actual and desired output, propagates in a backward direction starting from the output layer all the way back to the input layer. The back propagation algorithm is based on the minimization of the total error in a neural network between the actual and the desired output.

For the r-th training pattern of a network presented with p training patterns of inputs and outputs, the error for the k-th neuron of the output layer that is composed of y neurons is:

$$E(r)=1/2\left(\sum_{k=1}^{y}(d_k(r)-O_k(r))^2\right). \quad \text{(A-1)}$$

The total error (E_t) for all the patterns of the network is :

$$E_t=\sum_{k=1}^{y}E(r) \quad \text{(A-2)}$$

$$E_t=\frac{1}{2}\sum_{r=1}^{p}\sum_{k=1}^{y}(d_k(r)-O_k(r))^2 \quad \text{(A-3)}$$

Then all the weights among all the neurons are adjusted in order to minimize the total error.

$$\min(E_t)=\min\left(\frac{1}{2}\sum_{r=1}^{p}\sum_{k=1}^{v}(d_k(r)-O_k(r))^2\right) \quad \text{(A-4)}$$

$$\Delta w^s=-\zeta\frac{\partial E(r)}{\partial W^s} \quad \text{(A-5)}$$

Where s represents the number of the layer in the feed-forward network, Δw^s is the change in weight connection between $w^s(r)$ and $w^s(r+1)$ at layer (s) after presenting the subsequent training pattern (r+1).

ζ represents the learning rate which ranges from zero to one. And the term

$\frac{\partial E(r)}{\partial W^s}$ is the gradient of the error E(r) with respect to all the connection weights between layer (s) and the previous layer (s-1).

The technical difficulty to compute the gradient of the error E with respect to the connection weight w, is that there is not direct relation between them. The standard way of dealing

with this problem is to apply the chain rule to the equations to express the dependence of the output layer on the input layer.

$$\Delta w_{kj} = -\zeta \left[\frac{\partial E}{\partial O_k} \right] \left[\frac{\partial O_k}{\partial O_{ink}} \right] \left[\frac{\partial O_{ink}}{\partial w_{kj}} \right] \tag{A-6}$$

From equation (A-1) :

$$\frac{\partial E}{\partial O_k} = -(d_k(r) - O_k(r))$$

From equation (4-4) :

$$\frac{\partial O_k}{\partial O_{ink}} = a'(O_{ink}).$$

From equation (4-5) :

$$\frac{\partial O_{ink}}{\partial w_{kj}} = h_j .$$

Therefore the change in the connection weight w_{kj} between the j-th neuron in the hidden layer and the k-th neuron in the output layer is :

$$\Delta w_{kj} = \zeta (d_k - O_k) a'(O_{ink}) h_j . \tag{A-7}$$

$$\Delta w_{kj} = \zeta \alpha_k h_j . \tag{A-8}$$

where α_k is the error signal in the k-th output neuron and is represented as:

$$\alpha_k = (d_k - O_k) a'(O_{ink}). \tag{A-9}$$

Similarly the change in the connection weight u_{ji} between the i-th neuron in the input layer and the j-th neuron in the hidden layer is :

$$\Delta u_{ji} = -\zeta \left[\frac{\partial E}{\partial u_{ji}} \right] \tag{A-10}$$

$$\Delta u_{ji} = \left[\frac{\partial E}{\partial O_k} \right] \left[\frac{\partial O_k}{\partial O_{ink}} \right] \left[\frac{\partial O_{ink}}{\partial h_j} \right] \left[\frac{\partial h_j}{\partial u_{ji}} \right] \tag{A-11}$$

$$\Delta u_{ji} = \zeta \sum_{k=1}^{y} \left((d_k - O_k) a'(O_{ink}) w_{kj} a'(h_{inj}) m_i \right). \tag{A-12}$$

$$\Delta u_{ji} = \zeta \alpha_{ej} m_m. \tag{A-13}$$

where α_{ej} is the error signal in the j-th neuron of the e-th hidden layer and is represented as:

$$\alpha_{ej} = \sum_{k=1}^{y} \left((d_k - O_k) a'(O_{ink}) w_{kj} a'(h_{inj}) \right). \tag{A-14}$$

$$\alpha_{ej} = \sum_{k=1}^{y} a'(h_{inj}) \sum \alpha_k w_{kj}. \tag{A-15}$$

The error signals α_k and α_{ej} are propagated backward to update all the connection weights starting from the output layer all the way back to the first layer.

Some techniques were developed in order to speed up the convergence rate of the back propagation learning algorithm. As the back propagation algorithm demands infinitesimal steps. The utilization of small values of the learning rate could lead to a extremely slow convergence [79]. On the other hand, the utilization of large learning rate may cause unnecessary oscillations and may not cause convergence[163]. One of the ways to deal with this problem was proposed in ref [79],where adding a momentum term was proposed. It was shown that adding this term has led to a very fast convergence. Another way was proposed in reference [164], in which the authors have defined link metrics that allow for real time updates and only require minimum storage of information thus leading to fast and efficient implementation of the network.

Appendix B: Species-specific protein secondary structure prediction using orthologous datasets

B.1 Introduction

In this chapter, we will test hypothesis that, due to presence of species-specific differences in protein sequence and structure, we expect that protein secondary structure prediction methods trained on one species will outperform predictors trained on another species when orthologous protein datasets are used. Using datasets of orthologous proteins will highlight the species-specific elements of protein folding since other variables are held constant. For example, if we are to find the best classifier to predict human proteins, we expect that this classifier would be the one trained on human proteins and it would outperform other classifiers trained on orthologous proteins in other species. Results of this hypothesis failed due to the high degree of similarity in sequence and structure composition between orthologous proteins.

B.2 Methods

To test the hypothesis that a classifier trained solely on species A perform better on other proteins from species A then from a test dataset of orthologous proteins from species B, we sought to create species-specific datasets of orthologous proteins from multiple species. In order to remove confounding variables such as protein length, we first tried building species-specific datasets of proteins considered to be 'equivalent' between pairs of species. Such pairs of equivalent proteins are called orthologs. The concept of orthologs and the method of discovering orthologs is discussed below.

B.2.1 Detection of orthologs and inparalogs

Paralogs are originated after gene duplications. There are two different types of paralogs; the first type is called inparalogs and the second type is called outparalogs. Inparalogs originate through gene duplication process following speciation, while outparalogs originate after gene duplication process before speciation [165-168]. Orthologs can never be originated from outparalogs[166], nevertheless orthologs can be originated from inparalogs, as inparalogs are capable of forming orthologous genes with other species. Figure B-1 illustrates the concept of inparalogs, orthologs, and outparalogs. The figure shows that gene 'S' undergoes a gene duplication, then a speciation event occurs which gives rise to the two roots leading to species L and species K. The genes 'K2' and 'K3' are inparalogs as their gene duplication happened following the speciation event. Also the two genes 'K2' and 'K3' are co-orthologous to the gene 'L2' as they have one common ancestral protein. 'L1' is an outparalog of the genes 'K2' and 'k3'.

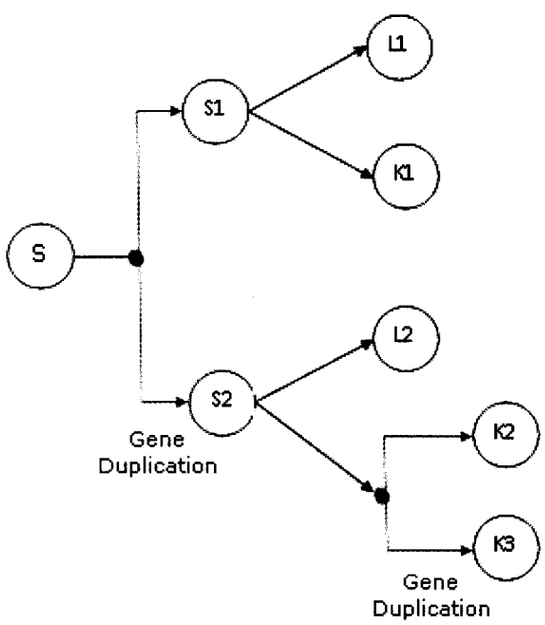

Figure B-1: Gene tree that illustrates inparalogs, orthologs, and outparalogs.

Orthologs and in-paralogs could be detected using phylogenetic analysis, but these methods are very slow and time consuming [166]. Some of the most popular ortholog identification methods are : COG (Clusters of Orthologous Groups) [170], KOG [171], InParanoid [166] , PGT [172] and OrthoMCL [173]. There have been several studies to compare different ortholog identification methods, it was found that the InParanoid program is the best ortholog identification method in terms of identifying functionally equivalent proteins [174][175][176].

Accordingly for this study, we have used the Inparanoid program [169] to identify these functionally equivalent inparalogs that have common ancestor in different species. The algorithm of the Inparanoid program allows the detection of orthologs from only two genomes at a time [176]. Figure B-2 shows a Venn diagram illustrating the subset of human and mouse proteins which form orthologous pairs. Note that for these two species, only approximately 10% of all proteins form orthologous pairs. The analysis in this chapter focuses on this small subset. The analysis in the next chapter uses all proteins from each species.

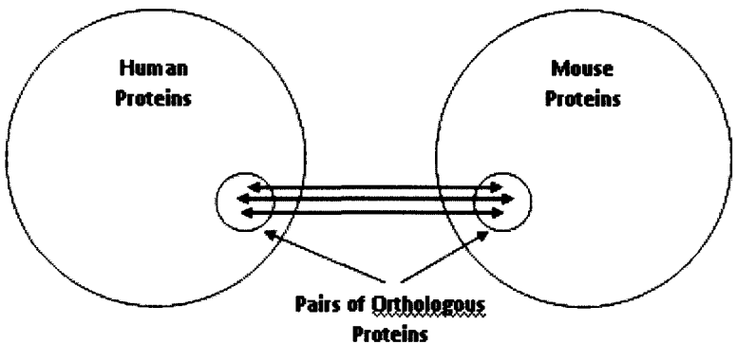

Figure B-2: Venn diagram showing subset of proteins which form orthologous pairs

B.2.2 Performance of species-specific Elman networks trained using orthologous proteins

To test the hypothesis that the optimum classifier for a certain species would be the classifier trained on proteins from same species, we have trained Elman networks on orthologous proteins between human and the other 11 species, also we have trained Elman networks on orthologous proteins between S. Cerevisiae and the other 11 species.

B.2.2.1 Preparation of datasets

We have downloaded a local version of the Inparanoid suite of PERL scripts [177] on the 15th of January 2008 to create training and testing datasets of functionally equivalent proteins (i.e. orthologs). As mentioned earlier, the Inparanoid program detects orthologs from only two species at a time. So, we ran Inparanoid on Human proteins vs. each of the other 11 species, one species at a time, then we ran Inparanoid on S. cerevisiae proteins vs. each of the other 11 species. The output of the Inparanoid program represents a number of ortholog groups, each ortholog group contains a number of (inparalogs) in each species species. A sample from Inparanoid is illustrated in Figure B-3.

```
Group of orthologs #1.

1ltk_B                  100.00%     3pgk_A                  92.90%
1ltk_A                  100.00%     1qpg_A                  100.00%
1ltk_C                  100.00%     1fw8_A                  54.32%
-------------------------------------------------------------------
Group of orthologs #2.

1onf_A                  100.00%     2hqm_A                  100.00%
                                    2hqm_B                  100.00%
-------------------------------------------------------------------
Group of orthologs #3.

1v0b_A                  100.00%     2pmi_A                  100.00%
1v0o_A                  100.00%     2pk9_C                  100.00%
1v0p_B                  100.00%     2pk9_A                  100.00%
1v0p_A                  100.00%     2pmi_C                  100.00%
1ob3_B                  100.00%
1ob3_A                  100.00%
1v0b_B                  100.00%
1v0o_B                  100.00%
```

Figure B-3: A sample of output of the Inparanoid program.

The figure shows three ortholog groups between S. Cerevisiae and P. Falciparum. Each ortholog group contains a number of inparalogs in each species and shows the corresponding bootstrapping score which is a measure of how reliably that protein is the main ortholog. We have chosen only one protein from each ortholog group to avoid redundancy of functionally equivalent proteins. The number of ortholog groups between human and the other 11 species is shown in Table B-1. As would be expected, the number of orthologs is high for species that are closely related to human (for example mouse), and low for distantly related species. There was no orthologs detected between human and P. Furiosus which indicates the absence of any orthologous proteins performing the same function in these two species. Also the ortholog groups between S. Cerevisaie and the other 11 species is shown in Table B-2.

Species	Ortholog groups
Human	444
Mouse	
Human	45
A. Thaliana	
Human	6
Z. Mays	
Human	11
A. Fulgidus	
Human	0
P. Furiosus	

Species	Ortholog groups
Human	49
B. Subtilis	
Human	42
T. Maritima	
Human	15
S. Pombe	
Human	189
S. Cerevisiae	
Human	32
P. Falciparum	
Human	19
T.Cruzi	

Table B-1: Number of orthologs between Human and the other 11 species.

Species	Ortholog groups
S. Cerevisiae / **Mouse**	56
S. Cerevisiae / ***A. Thaliana***	22
S. Cerevisiae / ***Z. Mays***	0
S. Cerevisiae / ***A. Fulgidus***	9
S. Cerevisiae / ***P. Furiosus***	4
S. Cerevisiae / ***B. Subtilis***	15
S. Cerevisiae / ***T. Maritima***	12
S. Cerevisiae / ***S. Pombe***	22
S. Cerevisiae / ***P. Falciparum***	11
S. Cerevisiae / ***T.Cruzi***	0

Table B-2: Number of orthologs between S. Cerevisiae and the other 11 species.

B.2 Results

B.2.1 Training Elman networks with functionally equivalent proteins across different species.

In order to test whether a classifier trained solely on one species would perform better on other proteins from same species than from a test dataset of orthologous proteins from another different species, we have used a five-fold cross validation to train Elman networks with orthologous proteins between human and the other 11 species, then S. Cerevisiae and the other 11 species. The results of training Elman networks with functionally equivalent

proteins between human and the other 11 species are illustrated in Table B-3. The results show that for some species, testing the classifier with proteins of the same species it was trained on has slightly improved the accuracy (with an average of only 0.38% improvement) when compared to testing the classifier with proteins of different species it was trained on . For example when species-specific Human and A. Thaliana datasets are used, training on the same species as the test set (e.g. train on Human, test on Human has accuracy 74.56%) provides slightly higher accuracy than training on a different species than the test dataset (e.g. train on Human, test on A. Thaliana has accuracy 74.23%). However, for some species (for example Human-Mouse, or Human- Z.Mays), there appears to be no consistent difference in accuracy when the same or different species are used for the training and testing.

Elman NN	Ortholog groups	Number of sequences	Train	Test	Accuracy
Human	444	78130	Human	Human	71.14
			Human	Mouse	71.25
Mouse		79435	Mouse	Mouse	71.86
			Mouse	Human	71.68
Human	45	9186	Human	Human	74.56
			Human	*A. Thaliana*	74.23
A. Thaliana		9173	*A. Thaliana*	*A. Thaliana*	74.59
			A. Thaliana	*Human*	74.46
Human	6	2112	Human	Human	66.25
			Human	*Z. Mays*	67.39
Z. Mays		1937	*Z. Mays*	*Z. Mays*	67.24
			Z. Mays	*Human*	67.13
Human	11	2736	Human	Human	70.95
			Human	*A. Fulgidus*	70.59
A. Fulgidus		2465	*A. Fulgidus*	*A. Fulgidus*	69.17
			A. Fulgidus	*Human*	71.12

Human	0				
P. Furiosus					
Human	49	12478	Human	Human	76.03
			Human	*B. Subtilis*	75.82
B. Subtilis		10401	*B. Subtilis*	*B. Subtilis*	75.4
			B. Subtilis	*Human*	75.94
Human	42	14182	Human	Human	73.56
			Human	*T. Maritima*	74.43
T. Maritima		12612	*T. Maritima*	*T. Maritima*	74.89
			T. Maritima	*Human*	74.41
Human	15	3149	Human	Human	68.84
			Human	*S. Pombe*	68.3
S. Pombe		3157	*S. Pombe*	*S. Pombe*	67.63
			S. Pombe	*Human*	66.79
Human	189	47470	Human	Human	74.73
			Human	*S. Cerevisiae*	73.93
S. Cerevisiae		47586	*S. Cerevisiae*	*S. Cerevisiae*	73.42
			S. Cerevisiae	*Human*	74.19
Human	32	7945	Human	Human	72.93
			Human	*P. Falciparum*	72.74
P. Falciparum		8321	*P. Falciparum*	*P. Falciparum*	72.9
			P. Falciparum	*Human*	73.25
Human	19	4988	Human	Human	73.66
			Human	*T.Cruzi*	73.32
T.Cruzi		5191	*T.Cruzi*	*T.Cruzi*	72.56
			T.Cruzi	*Human*	72.96

Table B-3: Training Elman networks with functionally equivalent proteins between human and other 11 species.

In order to support the results that training classifiers with species-specific orthologous proteins either leads to very slight improvement in the accuracy for some species or does not reflect any significant improvement in the accuracy for some other species when the classifier is tested with same or different species than it was trained on, we repeated training Elman networks with orthologous proteins between S. Cerevisiae and the other 11 species. The results are illustrated in Table B-4. The results again show that for some

species, testing the classifier with proteins of the same species it was trained on has resulted in a very slight improvement in the accuracy (with an average of 0.18% improvement) when compared to testing the classifier with orthologous proteins of different species it was trained on . For example when species-specific *S. Cerevisiae* and *A. Thaliana* datasets are used, training on *S. Cerevisiae*, testing on *S. Cerevisiae* has achieved an accuracy of 72.59% which is slightly higher than the accuracy achieved when training on *S. Cerevisiae*, testing on *A. Thaliana* which has achieved an accuracy of 72.57% (just 0.02% improvement in the accuracy). On the other hand, for some species (for example S. Cerevisiae-Mouse, *S. Cerevisiae-S. Pombe* or *S. Cerevisiae-Z. Mays*), there appears to be no consistent difference in accuracy when the same or different species are used for the training and testing.

Elman NN	Number of orthologs	Number of sequences	Train	Test	Accuracy
S. Cerevisiae	56	13513	*S. Cerevisiae*	*S. Cerevisiae*	73.83
			S. Cerevisiae	Mouse	74.66
Mouse			Mouse	Mouse	74.61
			Mouse	*S. Cerevisiae*	74.15
S. Cerevisiae	22	5042	*S. Cerevisiae*	*S. Cerevisiae*	72.59
			S. Cerevisiae	*A. Thaliana*	72.52
A. Thaliana			*A. Thaliana*	*A. Thaliana*	73.39
			A. Thaliana	*S. Cerevisiae*	73.61
S. Cerevisiae	0	0	*S. Cerevisiae*	*S. Cerevisiae*	
			S. Cerevisiae	*Z. Mays*	
Z. Mays			*Z. Mays*	*Z. Mays*	
			Z. Mays	*S. Cerevisiae*	
S. Cerevisiae	9	2132	*S. Cerevisiae*	*S. Cerevisiae*	71.92
			S. Cerevisiae	*A. Fulgidus*	71.87
A. Fulgidus			*A. Fulgidus*	*A. Fulgidus*	70.37
			A. Fulgidus	*S. Cerevisiae*	70.23
S. Cerevisiae	4	1176	*S. Cerevisiae*	*S. Cerevisiae*	62.47
			S. Cerevisiae	*P. Falciparum*	62.33
P. Furiosus			*P. Falciparum*	*P. Falciparum*	62.44
			P. Falciparum	*S. Cerevisiae*	62.65

S. Cerevisiae	15	3138	*S. Cerevisiae*	*S. Cerevisiae*	73.81
			S. Cerevisiae	*B. Subtilis*	73.84
B. Subtilis			*B. Subtilis*	*B. Subtilis*	73.84
			B. Subtilis	*S. Cerevisiae*	73.68
S. Cerevisiae	12	3952	*S. Cerevisiae*	*S. Cerevisiae*	64.63
			S. Cerevisiae	*T. Maritima*	65.49
T. Maritima			*T. Maritima*	*T. Maritima*	71.52
			T. Maritima	*S. Cerevisiae*	71.06
S. Cerevisiae	22	5042	*S. Cerevisiae*	*S. Cerevisiae*	72.74
			S. Cerevisiae	*S. Pombe*	72.96
S. Pombe			*S. Pombe*	*S. Pombe*	73.12
			S. Pombe	*S. Cerevisiae*	73.06
S. Cerevisiae	11	2790	*S. Cerevisiae*	*S. Cerevisiae*	61.29
			S. Cerevisiae	*P. Falciparum*	61.04
P. Falciparum			*P. Falciparum*	*P. Falciparum*	62.19
			P. Falciparum	*S. Cerevisiae*	62.09
S. Cerevisiae	0	0	*S. Cerevisiae*	*S. Cerevisiae*	
			S. Cerevisiae	*T.Cruzi*	
T.Cruzi			*T.Cruzi*	*T.Cruzi*	
			T.Cruzi	*S. Cerevisiae*	

Table B-4:Training Elman networks with functionally equivalent proteins between S. Cerevisiae and other 11 species.

B.2.2 Pearson chi-square test

We have used Pearson chi-square test [178] to assess the results shown in Table B-3and Table B-4, which show that there is no significant improvement in the accuracy when testing a classifier trained with species-specific than when training the classifier with orthologous proteins of another species which indicates the high degree of similarity between orthologous proteins in different species which consequently has a direct impact on the independence of the accuracy on the training and testing data of orthologous proteins. In order to quantitatively assess the independence of training and testing data, we have used the p-value to decide whether or not to reject the null hypothesis that the choice of training and testing datasets is independent. If the p-value is less than 0.05 , the null hypothesis can be rejected , otherwise the null hypothesis cannot be rejected. The results of using Pearson chi-square test are illustrated in Table B-5and Table B-6. Although one would expect that training Elman networks with orthologous species-specific datasets would have

an influence on improving the prediction accuracy, but the high p-values (an average of 0.9741 in Table B-5 and an average of 0.99 in Table B-6 of the chi-square test indicate that there is no significant difference in accuracy when the orthologous proteins of same or different species are used for the training and testing.

This may be due to the fact that proteins chosen by InParanoid are highly conserved between species indicating that their function is important to multiple species. These protein structures are likely to be highly conserved since they are apparently very important to all organisms and therefore the protein structures do not reflect the species-specific aspects of protein folding.

Elman NN	X^2 test
Human	0.997
Mouse	
Human	0.979
A. Thaliana	
Human	0.95
Z. Mays	
Human	0.924
A. Fulgidus	
Human	0.985
B. Subtilis	
Human	0.982
T. Maritima	
Human	0.935
S. Pombe	

Human	0.999
S. Cerevisiae	
Human	0.993
P. Falciparum	
Human	0.997
T.Cruzi	

Table B-5: Pearson chi-square test for orthologous proteins between human and other species.

Elman NN	X^2 test
S. Cerevisiae	0.983
Mouse	
S. Cerevisiae	0.993
A. Thaliana	
S. Cerevisiae	0.991
A. Fulgidus	
S. Cerevisiae	0.996
P. Falciparum	
S. Cerevisiae	0.994
B. Subtilis	
S. Cerevisiae	0.978
T. Maritima	
S. Cerevisiae	0.993
S. Pombe	
S. Cerevisiae	0.992
P. Falciparum	

Table B-6: Pearson chi-square test for orthologous proteins between S. Cerevisiae and other species.

B.3 Discussion

As shown in the previous section that there is no significant improvement in the accuracy when testing a classifier on same proteins it was trained than when training it with orthologous proteins from other species. I expect the main reason for that might be due the

presence of very high degree of similarity in the secondary structure and the sequence composition of orthologous proteins.

B.3.1 Secondary structure composition of orthologous proteins

One of the reasons that might lead to the absence of significant difference in accuracy when orthologous proteins of same or different species are used for the training and testing classifiers might be due to the presence of high structure conservation across orthologous proteins. We have created a Perl script to compute the percentage of secondary structure elements of orthologous proteins in human vs. the 11 other species. The results are shown in Table B-7. The results indicate the high degree of secondary structure similarity between orthologous proteins between human and the other 11 species. For example the percent of helices in orthologous proteins in human and mouse are 31.48% and 31.71% respectively with a very small difference of 0.33%.

	Percent of Helices	Percent of Beta-strands	Percent of coils	Paired T-test
Human	31.48	24.34	44.18	1
Mouse	31.71	24.35	43.94	
Human	37.95	20.13	41.91	0.999729
A. Thaliana	39.27	20.62	40.11	
Human	41.29	17.61	41.1	1
Z.Mays	38.98	18.43	42.59	
Human	42.36	19.37	38.27	1
A. Fulgidus	44.1	19.84	36.06	
Human	36.77	20.79	42.44	0.99972
B. Subtilis	38.09	21.14	40.76	
Human	40.31	19.59	40.1	0.999736
T.Maritima	41.28	20.39	38.34	
Human	34.55	20.8	44.65	1
S. Pombe	32.15	19.8	48.05	
Human	38.52	19.2	42.28	1
S. Cerevisiae	38.79	18.93	42.28	
Human	39.47	19.35	41.18	1
P. Falciparum	39.05	19.13	41.82	
Human	38.83	19.95	41.22	1
T. Cruzi	38.99	21.11	39.9	

Table B-7: Secondary structure composition of orthologous proteins.

To quantitavely assess the similarity in the secondary structure composition across orthologous proteins in human vs. the 11 other species, we have used a paired T-test . The results of the paired T-test shown in Table B-7. The results confirm that there is no significant difference in the secondary structure composition across the species, and these orthologous proteins have unusually high structure conservation. The secondary structure composition of orthologous proteins in human and mouse are illustrated in Figure B-4. The figure supports the hypothesis that orthologous proteins have unusually high structure conservation.

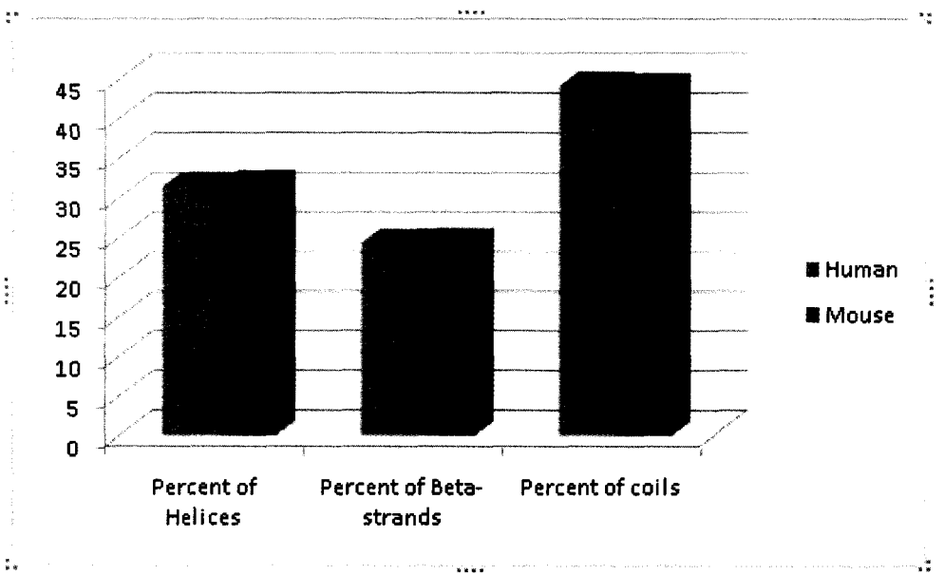

Figure B-4: Secondary structure composition of orthologs identified using Inparanoid in human vs. mouse.

B.3.2 Sequence composition of orthologous proteins

As has been shown that one of the main reasons that lead to the absence of significant difference in accuracy when orthologous proteins of same or different species are used for the training and testing classifiers is that these orthologous proteins have very similar secondary structure composition, so it has been of keen interest to investigate whether

these orthologous proteins have also similar sequence composition. Accordingly, we have created a Perl script to compute the percentage of sequence composition of orthologous proteins in human vs. the 11 other species. The results are illustrated in Table B-8. The results show the very high degree of similarity in sequence composition across orthologous proteins. We have used a paired T-test to assess the degree of sequence similarity of orthologs in human vs. the other 11 species. The results of the paired T-test are shown in Table B-8. The results indicate that there is no significant difference in the sequence composition across the species. The sequence composition of orthologs in Human and mouse are shown in Figure B-5. The figure points to the high degree of sequence similarity across orthologous proteins.

Species	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y	Paired T-test
Human	6.5	1.88	5.39	6.91	4.02	7.02	2.87	4.85	6.39	9.1	2.22	4.1	4.82	4.32	4.92	7.71	5.36	6.81	1.45	3.34	0.998160155
Mouse	6.5	1.9	5.41	6.76	4.04	7.15	2.73	4.84	6.37	8.98	2.26	4.05	4.83	4.39	4.88	7.89	5.42	6.75	1.42	3.38	
Human	7.29	1.88	5.52	7.32	3.95	7.78	2.78	5.39	6.6	9.17	2.73	3.64	4.82	4.02	4.72	6.37	4.99	6.97	1.06	2.98	0.999440323
A. Thaliana	8.25	1.46	5.77	7.16	4.32	7.69	2.35	5.83	6.54	8.42	2.31	3.79	4.54	3.22	4.33	7.48	5.42	7.29	1.02	2.8	
Human	6.78	1.08	5.96	6.19	4.52	6.5	2.71	5.74	6.23	10.3	2.03	3.43	5.83	4.87	5.37	5.28	4.87	6.59	1.76	3.97	1
Z. Mays	6.94	0.83	6.67	6.47	3.97	6.92	2.65	5.05	6.52	9.22	2.4	4.46	5.69	3.19	5.2	5.35	4.12	7.3	1.57	5.44	
Human	7.49	1.68	4.82	7.42	3.42	7.8	3.22	5.2	6.6	9.68	2.94	3.97	4.65	4.32	3.8	6.53	4.55	8.14	0.89	2.87	0.999515933
A.Fulgidus	8.48	0.83	5.73	9.65	3.7	7.13	1.92	7.13	6.86	9.62	3.24	3.32	3.7	1.47	6.26	5.39	3.54	8.22	0.87	2.94	
Human	7.55	1.94	5.34	6.17	3.92	7.85	3.32	5.51	5.73	9.24	2.59	3.76	5.03	4.07	4.73	6.43	5.06	7.34	1.25	3.14	1
B. Subtilis	8.57	0.91	5.91	7.76	3.84	7.85	2.66	6.52	6.3	8.47	2.32	4.28	4.24	3.6	4.08	6.03	5.19	6.87	1.09	3.48	
Human	8.15	1.82	5.38	6.73	4.01	7.69	2.68	5.52	6.33	9.78	2.4	3.34	4.54	3.9	4.83	6.35	5.2	7.48	1.07	2.78	0.999001734
T. Maritima	7.1	0.72	5.33	9.22	4.36	7.98	2.8	6.88	7.74	9.06	2.59	3.41	3.85	1.7	5.13	5.04	4.52	8.76	0.88	2.93	
Human	6.17	1.97	5.99	7.72	4.55	7.02	2.94	5.96	7.78	8.63	2.38	3.88	4.67	3.85	4.32	5.52	4.87	6.4	1.47	3.88	0.998777653
S. Pombe	6.8	1.49	6.07	7.26	4.48	6.33	1.96	6.13	6.57	8.97	2.05	4.57	4.69	3.46	5.07	6.89	5.07	6.62	1.29	4.22	
Human	7.12	1.77	5.57	7.22	4.15	6.92	2.76	5.49	6.75	9.56	2.53	3.93	4.69	4.17	4.85	6.29	5.03	6.8	1.25	3.18	0.998235996
S.Cerevisiae	6.61	1.27	6.33	6.9	4.55	6.22	2.59	6.31	7.35	9.2	2.09	5.01	4.42	3.71	4.09	6.98	5.26	6.4	1.11	3.6	
Human	8.09	1.58	5.31	5.99	3.84	8.8	2.57	5.7	6.74	9.15	2.36	3.91	4.92	3.68	4.85	5.96	4.74	7.87	1.07	2.86	0.999479436
P. Falciparum	4.94	1.78	5.78	6.45	4.6	6.71	2.94	7.81	9.2	9.32	1.94	7.68	3.11	2.43	3.05	5.85	4.75	6.62	0.87	4.15	
Human	7.81	1.58	5.69	5.95	3.78	9.21	2.73	5.59	6.55	8.85	2.37	3.69	4.93	3.67	4.46	5.78	5.2	8.12	1.07	2.96	0.999461691
T. Cruzi	8.92	1.83	5.24	6.62	4.17	8.31	2.32	4.59	5.89	8.58	2.63	3.94	4.7	3.06	5.06	5.73	5.53	8.49	1.11	3.28	

Table B-8: Sequence composition of orthologous proteins.

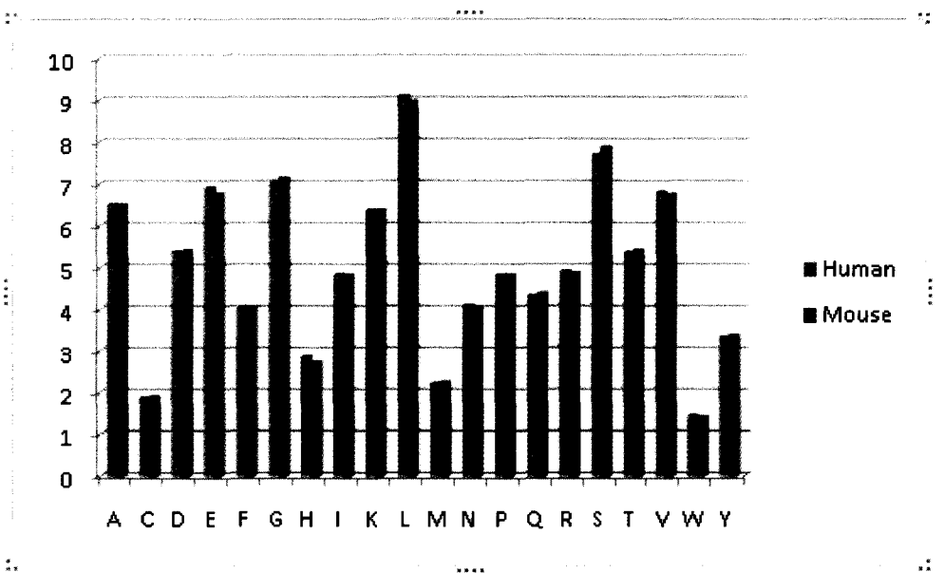

Figure B-5: Sequence composition of orthologs identified using Inparanoid in Human vs. mouse.

B.4 Conclusions

In this chapter, we have tested the hypothesis that, due to species-specific aspects of protein folding, we have expected there to be an advantage to training structure prediction methods using species-specific datasets of orthologous proteins. But the results presented in this chapter indicated that there is no significant improvement in the accuracy when testing classifiers on proteins of same species it was trained on than when training classifier on orthologous proteins of different species. It has been shown that these orthologous proteins have unusually high secondary structure and sequence conservation which has lead to the absence of significant difference in accuracy when orthologous proteins of same or different species are used for the training and testing classifiers. Accordingly, we decided to change experiment design to not limit analysis to orthologous proteins.

Appendix C: Detailed results of species-specific prediction accuracies

Appendix C includes detailed results of testing 13 classifiers (trained on the 12 species that are used throughout this thesis plus one classifier trained on mixed species) on 16 new datasets to find the optimum classifier for each of these new 16 species. This optimum classifier for the 16 species is found to be the one trained on other phylogenetically similar species in the same Kingdom or from closely related Kingdoms.

Kingdom	Species	Accuracy
Animals	Human	66.83
	Mouse	**67.67**
Plants	*A. Thaliana*	66.63
	Z. Mays	66.42
Archaebacteria	*A. Fulgidus*	67.1
	P. Furiosus	66.36
Eubacteria	*B. Subtilis*	66.07
	T. Maritima	66.86
Fungi	*S. Pombe*	67.35
	S. Cerevisiae	67.17
Protists	*P. Falciparum*	67.22
	T. Cruzi	67.35
Mixed_species		66.55

Table C-1: Testing on Bos Taurus (Animal Kingdom).

Kingdom	Species	Accuracy
Animals	Human	68.03
	Mouse	**68.97**
Plants	*A. Thaliana*	68.68
	Z. Mays	68.27
Archaebacteria	*A. Fulgidus*	68.21
	P. Furiosus	68.55
Eubacteria	*B. Subtilis*	67.88
	T. Maritima	68
Fungi	*S. Pombe*	68.4
	S. Cerevisiae	66.25
Protists	*P. Falciparum*	67.3
	T. Cruzi	68.4
Mixed_species		67.57

Table C-2: Testing on Rattus Norvegicus (Animal Kingdom).

Kingdom	Species	Accuracy
Animals	Human	67.87
	Mouse	67.82
Plants	***A. Thaliana***	**68.22**
	Z. Mays	67.2
Archaebacteria	*A. Fulgidus*	67.76
	P. Furiosus	67.41
Eubacteria	*B. Subtilis*	66.71
	T. Maritima	67.29
Fungi	*S. Pombe*	68.11
	S. Cerevisiae	66.97
Protists	*P. Falciparum*	66.81
	T. Cruzi	68.11
Mixed_species		67.41

Table C-3: Testing on Drosophila Melanogaster (Animal Kingdom).

Kingdom	Species	Accuracy
Animals	**Human**	**67.15**
	Mouse	65.54
Plants	*A. Thaliana*	64
	Z. Mays	60.71
Archaebacteria	*A. Fulgidus*	60.25
	P. Furiosus	64.43
Eubacteria	*B. Subtilis*	56.61
	T. Maritima	63.92
Fungi	*S. Pombe*	65.68
	S. Cerevisiae	51.39
Protists	*P. Falciparum*	52.75
	T. Cruzi	65.68
Mixed_species		64.19

Table C-4: Testing on *Canavalia Ensiformis* (Plant Kingdom).

Kingdom	Species	Accuracy
Animals	Human	62.2
	Mouse	64.07
Plants	***A. Thaliana***	**64.32**
	Z. Mays	63.04
Archaebacteria	*A. Fulgidus*	57.2
	P. Furiosus	63.39
Eubacteria	*B. Subtilis*	63.11
	T. Maritima	61.2
Fungi	*S. Pombe*	61.4
	S. Cerevisiae	59.17
Protists	*P. Falciparum*	61.42
	T. Cruzi	61.4
Mixed_species		63.08

Table C-5: Testing on *Glycine Max* (Plant Kingdom).

Kingdom	Species	Accuracy
Animals	Human	69.31
	Mouse	70.16
Plants	*A. Thaliana*	70.8
	Z. Mays	70.79
Archaebacteria	*A. Fulgidus*	70.6
	P. Furiosus	69.97
Eubacteria	*B. Subtilis*	69.65
	T. Maritima	70.02
Fungi	***S. Pombe***	**71.06**
	S. Cerevisiae	68.63
Protists	*P. Falciparum*	69.55
	T. Cruzi	70.46
Mixed_species		70.17

Table C-6: Testing on *Pisum Sativum* (Plant Kingdom).

Kingdom	Species	Accuracy
Animals	Human	59.23
	Mouse	56.59
Plants	*A. Thaliana*	62.35
	Z. Mays	62.55
Archaebacteria	***A. Fulgidus***	**70.52**
	P. Furiosus	62.98
Eubacteria	*B. Subtilis*	64.45
	T. Maritima	64.34
Fungi	*S. Pombe*	63.31
	S. Cerevisiae	67.64
Protists	*P. Falciparum*	70.5
	T. Cruzi	63.31
Mixed_species		63.32

Table C-7: Testing on *Halobacterium Salinarum* (Archaebacteria Kingdom).

Kingdom	Species	Accuracy
Animals	Human	68.56
	Mouse	69.31
Plants	*A. Thaliana*	69.55
	Z. Mays	67.89
Archaebacteria	***A. Fulgidus***	**70.39**
	P. Furiosus	69.85
Eubacteria	*B. Subtilis*	69.03
	T. Maritima	69.67
Fungi	*S. Pombe*	68.32
	S. Cerevisiae	66.8
Protists	*P. Falciparum*	67.95
	T. Cruzi	68.12
Mixed_species		69.67

Table C-8: Testing on *Methanocaldococcus Jannaschit* (Archaebacteria Kingdom).

Kingdom	Species	Accuracy
Animals	Human	70.61
	Mouse	70.08
Plants	*A. Thaliana*	71.66
	Z. Mays	70.98
Archaebacteria	***A. Fulgidus***	**72.81**
	P. Furiosus	72.04
Eubacteria	*B. Subtilis*	71.56
	T. Maritima	72.59
Fungi	*S. Pombe*	71.75
	S. Cerevisiae	69.51
Protists	*P. Falciparum*	70.93
	T. Cruzi	71.75
Mixed_species		71.97

Table C-9: Testing on *Sulfolobus Solfataricus* (Archaebacteria Kingdom).

Kingdom	Species	Accuracy
Animals	Human	70.99
	Mouse	71.21
Plants	*A. Thaliana*	71.7
	Z. Mays	69.86
Archaebacteria	*A. Fulgidus*	70.48
	P. Furiosus	**72.45**
Eubacteria	*B. Subtilis*	70.89
	T. Maritima	71.57
Fungi	*S. Pombe*	70.65
	S. Cerevisiae	70.2
Protists	*P. Falciparum*	69.23
	T. Cruzi	70.65
Mixed_species		71.1

Table C-10: Testing on *Thermoplasma Acidophilum* (Archaebacteria).

Kingdom	Species	Accuracy
Animals	Human	69.23
	Mouse	69.98
Plants	*A. Thaliana*	70.1
	Z. Mays	70.79
Archaebacteria	*A. Fulgidus*	71.34
	P. Furiosus	70.81
Eubacteria	***B. Subtilis***	**72.07**
	T. Maritima	71.8
Fungi	*S. Pombe*	70.71
	S. Cerevisiae	69.66
Protists	*P. Falciparum*	70.79
	T. Cruzi	71.12
Mixed_species		71.21

Table C-11: Testing on *Aquifex Aeolicus* (Eubacteria Kingdom).

Kingdom	Species	Accuracy
Animals	Human	68.3
	Mouse	69.03
Plants	*A. Thaliana*	69.78
	Z. Mays	67.99
Archaebacteria	*A. Fulgidus*	70.1
	P. Furiosus	68.54
Eubacteria	*B. Subtilis*	69.72
	T. Maritima	**70.13**
Fungi	*S. Pombe*	68.83
	S. Cerevisiae	67.89
Protists	*P. Falciparum*	67.7
	T. Cruzi	68.83
Mixed_species		69.59

Table C-12: Testing on *Haemophilus Influenzae* (Eubacteria Kingdom).

Kingdom	Species	Accuracy
Animals	Human	73.82
	Mouse	72.65
Plants	*A. Thaliana*	72.61
	Z. Mays	73.78
Archaebacteria	*A. Fulgidus*	72.09
	P. Furiosus	73.52
Eubacteria	*B. Subtilis*	69.18
	T. Maritima	71.89
Fungi	*S. Pombe*	73.2
	S. Cerevisiae	**73.96**
Protists	*P. Falciparum*	69.35
	T. Cruzi	71.65
Mixed_species		68.94

Table C-13: Testing on *Aspergillus Oryzae* (Fungi Kingdom).

Kingdom	Species	Accuracy
Animals	Human	67.66
	Mouse	67.57
Plants	*A. Thaliana*	68.12
	Z. Mays	66.83
Archaebacteria	*A. Fulgidus*	67.92
	P. Furiosus	67.77
Eubacteria	*B. Subtilis*	67.71
	T. Maritima	68.11
Fungi	***S. Pombe***	**68.74**
	S. Cerevisiae	66.17
Protists	*P. Falciparum*	65.1
	T. Cruzi	68.1
Mixed_species		67.73

Table C-14: Testing on *Candida Albicans* (Fungi Kingdom).

Kingdom	Species	Accuracy
Animals	Human	66.33
	Mouse	68.21
Plants	*A. Thaliana*	67.01
	Z. Mays	67.11
Archaebacteria	*A. Fulgidus*	67.08
	P. Furiosus	66.1
Eubacteria	*B. Subtilis*	68.13
	T. Maritima	67.21
Fungi	*S. Pombe*	68.26
	S. Cerevisiae	66.11
Protists	*P. Falciparum*	68.1
	T. Cruzi	**68.93**
Mixed_species		66.86

Table C-15: Testing on *Dictyostelium Discodeum* (Protist Kingdom).

References

1. B. Rost, "Review: protein secondary structure prediction continues to rise", J Struct Biol., vol. 134, no. 2-3, pp. 204-218, 2001.
2. Y. Li, H. Zhou, L. Wang, and J. Liu, "Predicting Protein Secondary Structure by a Support Vector Machine Based on a New Coding Scheme", Genome Informatics, vol. 15, no. 2, pp. 181–190, 2004.
3. J. Garnier, D.J. Osguthorpe,and B. Robson, "Analysis of the accuracy and implication of simple methods for predicting the secondary structure of globular proteins", J. Mol. Biol., vol. 120, pp. 97-120, 1978.
4. A. Kloczkowski, K. L. Ting, R. L. Jernigan and J. Garnier J., " Protein secondary structure prediction based on the GOR algorithm incorporating multiple sequence alignment information", Polymer, vol. 43, no. 2, pp. 441-449, 2002.
5. N. Qian, and T.J. Sejnowski, "Predicting the secondary structure of globular proteins using neural network models", J. Mol. Biol., vol. 202, pp. 865-884, 1988.
6. B. Rost, B., Sander, C. "Prediction of protein secondary structure at better than 70% accuracy" J. Mol. Biol., vol. 232, pp. 584-599, 1993.
7. D.T. Jones, "Protein Secondary Structure Prediction Based on Position-specific Scoring Matrices", J. Mol. Biol., vol. 292, pp. 195-202, 1999.
8. J. Chen and N.S. Chaudhari, "Bidirectional segmented-memory recurrent neural network for protein secondary structure prediction" Soft Comput, vol. 10, pp. 315-324, 2006.
9. T. Yi, and E. Lander, "Protein secondary structure prediction using nearest-neighbour methods " , J Mol Biol., vol 232, no. 4, pp. 1117-1129, 1993.
10. K. Bryson, L. J. McGuffin, R. L. Marsden, J. J. Ward, J. S. Sodhi and D. T. Jones, " Protein structure prediction servers at University College London", Nucleic Acids Research , vol. 33, pp. 36-38, 2005.

11. D. T. Jones, " 12 Years a CASP predictor: A look back and a look ahead", http://cubic.bioc.columbia.edu/meetings/casp-6-5/slides/casp65_jones.pdf
12. D. Amaratunga and J. Cabrera, Exploration and Analysis of DNA Microarray and Protein Array Data, Wiley-IEEE Press, 2004.
13. Raincoast B., *The Encyclopedic Atlas of the Human Body*. Raincoast Books Press, 2004.
14. E. Tamir, *The Human Body Made Simple: A Guide to Anatomy, Physiology, and Disease*. Elsevier Health Sciences Press, 2002.
15. M. K. Ganapathiraju, J. Klein-Seetharaman, N. Balakrishnan, and R. Reddy, "Characterization of protein secondary structure" , Signal Processing Magazine, IEEE, vol. 21, no. 3, pp. 78-87, 2004.
16. S. Yoon, "Technologies and Analysis Methods for Detecting Gene Expression by DNA Microarrays", IEEE Technology Surveys, 2006.
17. A. M. Lesk, *Introduction to Bioinformatics*, Oxford University Press, 2005.
18. K. U. Linderstrom-Lang and J. A. Schellman, *Protein structure and enzyme activity in the Enzymes*, New York: Academic Press, 1959.
19. M. Lesk, *Introduction to Genomics*, Oxford University Press, 2007.
20. I. Klotz, D. Darnall, and N. Langerman, " Quaternary structure of proteins", The Proteins, vol. 1, pp. 293–411, 1975.
21. Characteristics and Properties of Amino Acids, http://www.elmhurst.edu/~chm/vchembook/561aminostructure.html.
22. A Review of Amino Acids, http://www.biomed.curtin.edu.au/biochem/tutorials/AAs/AA.html.
23. A. Procopiou,N. M. Allinson,G. R. Jones, and D. T. Clarke, "Estimation of protein secondary structure from synchrotron radiation circular dichroism spectra" , Engineering in Medicine and Biology Society, IEMBS '04. 26th

Annual International Conference of the IEEE, vol.2, pp. 2893–2896, 2004.

24. W. Kabsch, and C. Sander, "Dictionary of Protein Secondary Structure: Patt ern Recognition of Hydrogen-Bonded and Geometrical Features", Biopolymers, vol. 22, pp. 2577-2637, 1983.
25. T. E. Creighton, *Proteins: Structures and Molecular Properties*. New York, W. H. Freeman Press, 1993.
26. C. M. Venkatachalam, "Stereochemical criteria for polypeptides and proteins. V. Conformation of a system of three linked peptide units", Biopolymers, vol. 6, pp. 1425–1436, 1968.
27. C. Branden and J. Tooze, *Introduction to Protein Structure*. Garland Press, 1999.
28. A. Gregory, Petsko, R. Dagmar, *Protein Structure and Function*. New Science Press, 2004.
29. G. Fasman, *Prediction of protein structure and the principles of protein conformation.* Plenum Press, 1989.
30. M. K. Ganapathiraju, J. Klein-Seetharaman, N. Balakrishnan, and R. Reddy, "Characterization of protein secondary structure" , Signal Processing Magazine, IEEE, vol. 21, no. 3, pp. 78- 87, 2004.
31. G. Rhodes, *Crystallography Made Crystal Clear: A Guide for Users of Macromolecular Models*. Academic Press, 2006.
32. H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The Protein Data Bank", Nucleic Acids Research, vol. 28, pp. 235-242, 2000.
33. I. Jurisica, P. Rogers, J. Glasgow, S. Fortier, J. Luft, J. Wolfley, M. Bianca, D. Weeks, and G. DeTitta, "Intelligent Decision Support for Crystal Growth", IBM Systems Journal, vol. 40, no. 2, 2001.

34. W. Kurt, "NMR studies of structure and function of biological macromolecules", J Biomol NMR, vol. 27, no. 1, pp. 13-39, 2003.
35. T. E. Creighton, *Proteins: Structures and Molecular Properties*. W.H. Freeman & Company Press, 1992.
36. M. Sattler and B. Simon, "NMR of large proteins" , e-magazine of the European Life Scientist Organization, vol. 11, 2002.
37. G. Wider, "High-resolution nuclear magnetic resonance applied to biophysics and molecular biology: highlights and challenges", Applied Superconductivity, IEEE Transactions, vol. 12, no. 1, pp. 740 – 745, 2002.
38. A. Horwich, and K. Wüthrich, "NMR analysis of a 900K GroEL–GroES complex", Nature, vol. 418, pp. 207-211, 2002.
39. R. Kumar, S. Pavithra, and U. Tatu, "Three-dimensional structure of heat shock protein 90 from Plasmodium falciparum: molecular modelling approach to rational drug design against malaria", J. Biosci., vol. 32, no. 3, 2007.
40. C.J. Epstain, R.F. Goldberger, and C.B. Anfinsen, "The Genetic Control of Tertiary Protein Structure: Studies With Model Systems", Cold Spring Harb Symp Quant Biol, vol. 1, no. 28, pp. 439, 1963.
41. D. Gerald, *Prediction of Protein Structure and the Principles of Protein Conformation*. Springer Press, 1989.
42. E. Krieger, S. Nabuurs, and G. Vriend, "Homology modeling" , Methods of biochemical analysis, vol. 44, pp. 509-23, 2003.
43. A. Gregory, and R. Dagmar, *Protein Structure and Function*. New Science Press, 2004.
44. W. Yong, W. Ling-Yun, Z. Xiang-Sun, and C. Luonan, Protein Comparison Based on Both Structure and Sequence Data. Springer Berlin Heidelberg,

2007.

45. A. Morris, M. MacArthur, E. Hutchinson, and J. Thornton, "Stereochemical quality of protein structure coordinates" , Proteins, vol. 12, pp. 345-364, 1992.

46. C. Chothia, and A. M. Lesk, "The relation between the divergence of sequence and structure in proteins" , EMBO J., vol. 5, pp. 823–826, 1986.

47. R. F. Doolittle, "Similar amino acid sequences: chance or common ancestry?", Science, vol. 214, PP. 149–159, 1981.

48. C. Sander, and R. Schneider, "Database of homology-derived structures and the structural meaning of sequence alignment" , Proteins, vol. 9, PP. 56-68, 1991.

49. B. Rost , "Twilight zone of protein sequence alignments ", Protein Engineering, vol. 12, pp. 85-94, 1999.

50. A. Fiser, R. K. Do, and A. Sali "Modeling of loops in protein structures" , Protein science, vol. 9, pp. 1753–1773, 2000.

51. F. András, and S. Andrej," ModLoop: automated modeling of loops in protein structures", Bioinformatics, vol. 19, pp. 2500–2501, 2003.

52. K. Simons, I. Ruczinski, C. Kooperberg, B. Fox, C. Bystroff, and D. Baker, "Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence- independent features of proteins" , Proteins, vol. 34,pp. 82–95, 1999.

53. S. Altschul, T. Madden, A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. Lipman, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs" , Nucleic Acids Res., vol. 25, pp. 3389-3402, 1997.

54. S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic

local alignment search tool." J. Mol. Biol., vol. 215, pp. 403-410, 1990.

55. A. Lesk, *Introduction to Bioinformatics.* Oxford university Press, 2006.

56. M. A. Martí-Renom, A. C. Stuart, A. Fiser, R. Sánchez, F. Melo, and A. Šali, "Comparative protein structure modeling of genes and genomes", Annual review of biophysics and biomolecular structure, vol. 29, pp. 291-325, 2000.

57. B. Rost, R. Schneider, and C. Sander, "Protein fold recognition by prediction-based threading", J Mol Biol., vol. 270, pp. 471-480, 1997.

58. A. Tramontano, Introduction to Bioinformatics. Crc Press, 2007.

59. E. Tafeit , W. Estelberger, R. Horejsi, R. Moeller, K. Oettl, K. Vrecko, and G. Reibnegger, "Neural networks as a tool for compact representation of ab initio molecular potential energy surfaces" , J Mol Graph, vol. 14, no. 1, pp. 12-18, 1996.

60. I. Chang, M. Cieplak, R. Dima, A. Maritan, and J. Banavar, "Protein threading by learning ", Proc Natl Acad Science, vol. 98, no. 25, pp. 14350–14355, 2001.

61. C. Anfinsen, "Principles that govern the folding of protein chains", Science, vol. 181, no. 96, pp. 223-230, 1973.

62. D. Jones, "Progress in protein structure prediction", Curr Opin Struct Biol., vol. 7, no. 3, pp. 377-387, 1997.

63. L. Adam, L. Jooyoung, R. Daniel, P. Jaroslaw, and A. Harold A. "Protein structure prediction by global optimization of a potential energy function" , PNAS, vol. 96, no. 10, pp. 5482-5485, 1999.

64. P. Jarosław, C. Cezary, L. Adam, L. Jooyoung, R. Daniel, K. Rajmund, O. Stanisław, J. William, D. Kenneth, A. Yelena, S. Jeff, Y. Yuan-Jie, and A. Harold, "Recent improvements in prediction of protein structure by global

optimization of a potential energy function" , Proc Natl Acad Science, vol. 98, no. 5, pp. 2329–2333, 2001.

65. H. Scheraga, "Recent developments in the theory of protein folding: searching for the global energy minimum", Biophys Chem., vol 59, pp. 329–339, 1996.

66. M. Hoque, M. Chetty, and L. S. Dooley, *Significance of Hybrid Evolutionary Computation for Ab Initio Protein Folding Prediction*. 2007.

67. X. Wang, D. Schroeder, D. Dobbs, and V. Honavar, "Automated data-driven discovery of motif-based protein function classifiers " , Information Sciences , vol. 155, pp. 1-18, 2003.

68. J. Klepeis, and C. Floudas, " ASTRO-FOLD: a combinatorial and global optimization framework for ab initio prediction of three dimensional structures of proteins from the amino acid sequence" Biophysical Journal, vol.85, pp. 2119-2146, 2003.

69. B. Rost B, "Review: protein secondary structure prediction continues to rise", J Struct Biol., vol. 134, no. 2-3, pp. 204-218, 2001.

70. Z. Aydin, and Y. Altunbasak, "A signal processing application in genomic research: protein secondary structure prediction", Signal Processing Magazine, IEEE, vol. 23, no. 4, pp. 128–131, 2006.

71. C. Zhang, J. T. Hou, and S. H. Kim, "Fold prediction of helical proteins using torsion angle dynamics and predicted restraints," Proc. Nat. Acad. Science, vol. 99, pp. 3581–3585, 2002.

72. P. Munson, and J. Garnier, "FORESST: fold recognition from secondary structure predictions of proteins", Bioinformatics, vol 15, pp. 131-140, 1999.

73. H. Cheng, and R. Jerniga "Prediction of protein secondary structure by

mining structural fragment database", Polymer, vol. 46, no.12, pp. 4314-4321, 2005.

74. Y. Li, H. Zhou, L. Wang, and J. Liu, " Predicting Protein Secondary Structure by a Support Vector Machine Based on a New Coding Scheme " Genome Informatics, vol 15(2): 181–190, 2004.
75. N. Qian, and T.J. Sejnowski, "Predicting the secondary structure of globular proteins using neural network models", J. Mol. Biol., vol. 202, pp. 865-884, 1988.
76. H. L. Holley, and M. Karplus, "Protein secondary structure prediction with a neural network", Proc. Natl. Acad. Sci., vol. 86, pp. 152-156, 1989
77. Introduction to neural networks, http://www.ii.metu.edu.tr/~ion526/demo/chapter1/section1.1/index.html
78. J. Heaton, "Introduction to neural networks with Java", Heaton research Press, 2005.
79. k. Fakhareddine, and D. Clarence, *Soft computing and intelligent system design.* Addison Wesley Press, 2008.
80. P. Werbos, "Beyond regression: New tool for prediction and analysis in the behavioral sciences", Ph.D. Thesis, Cambridge, MA: Harvard University, 1974.
81. B. Widrow and S. Stearns, "Adaptive signal processing", Englewood Cliffs, NJ, Prentice Hall, 1985.
82. L. R. Medsker, and L. C. Jain, *Recurrent Neural Networks: Design and Applications*. CRC Press, 2000.
83. H. Song, S. Kang, and S. Lee, "A new recurrent neural network architecture for pattern recognition", IEEE Proceedings of the 13th International Conference on pattern recognition, vol. 4, pp. 718-722, 1996.

84. W. S. McCulloh, W. and Pitts, "A logical calculus of the ideas immanent in nervous activity" , Bulletin of Mathematical Biophysics, vol. 5, pp. 115, 1943 .

85. W. Pitts, and W. S. McCulloh, "How we know universals", Bulletin of Mathematical Biophysics, vol. 9, pp. 127, 1947.

86. D. Garson, *Neural networks: An introductory guide for social scientists.* Sage Press, 1998.

87. F. Rosenblatt, "The Perceptron: A probalistic model for information storage and organisation in the brain", Physiological review, vol. 65, pp. 386-408, 1958.

88. B. Reinhardt, *Neural networks.* Springer Verlag Press, 1991.

89. M. Minsky, and S. Papert*, Perceptrons: an introduction to computational geometry*. MIT Press. 1969.

90. W. Thomas, M. Thomas, P. Werbos, and R. Sutton, Neural Networks for Control. MIT Press, 1995.

91. Chou P. Y. and Fasman G. D., "Prediction of secondary structure of proteins from their amino acid", Adv. Enzymol, vol. 47, pp. 45-148, 1978.

92. Lim V. I., " Algorithms for prediction of alpha-helical and beta-structural regions in globular proteins.", J Mol Biol. vol. 88, no. 4, pp. 873–894, 1974.

93. B. Rost, B., Sander, C. "Prediction of protein secondary structure at better than 70% accuracy" J. Mol. Biol., vol. 232, pp. 584-599, 1993.

94. D.T. Jones, "Protein Secondary Structure Prediction Based on Position-specific Scoring Matrices", J. Mol. Biol., vol. 292, pp. 195-202, 1999.

95. Altschul S. F., Madden T. L., Schäffer A. A., Zhang J., Zhang Z., Miller W.and Lipman D. J. ," Gapped BLAST and PSI-BLAST: a new generation of protein database search programs" , Nucleic Acids Res., vol. 25, no. 17, pp.

3389–3402, 1997.

96. Bryson K., McGuffin L.J., Marsden R. L., Ward J. J., Sodhi J. S. and Jones D. T. "Protein structure prediction servers at university college London", Nucleic Acid Research, vol. 33, 2005.

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

98. J. Cuff, and G. Barton, "Evaluation and improvement of multiple sequence methods for protein secondary structure prediction" Proteins, vol. 34, pp. 508–519, 1999.

99. M. Chandonia, M. Karplus, "New methods for accurate prediction of protein secondary structure", Proteins, vol. 35, pp. 293–306, 1999.

100. T.N. Petersen, C. Lundegaard, M. Nielsen, H. Bohr, J. Bohr, S. Brunak, GP. Gippert, and O. Lund, "Prediction of protein secondary structure at 80% accuracy", Proteins, vol. 41, pp. 17-20, 2000.

101. J. Chen and N.S. Chaudhari, "Bidirectional segmented-memory recurrent neural network for protein secondary structure prediction" Soft Comput, vol. 10, pp. 315-324, 2006.

102. C. Jinmiao, and N.S. Chaudhari, "Improvement of bidirectional recurrent neural network for learning long-term dependencies", 17th IEEE International Conference, vol. 4, pp. 593–596, 2004.

103. J. Chen and N.S. Chaudhari, "Capturing Long-Term Dependencies for Protein Secondary Structure Prediction" SpringerLink, vol. 3174, pp. 494-500, 2004.

104. J. Seoane, L. M. Carrascal , C. L. Alonso, and D. Palomino, "Species-specific traits associated to prediction errors in bird habitat suitability

modeling", Ecological Modeling, vol. 185, pp. 299-308, 2005.

105. R. MacNally, and A. F. Bennett, " Species-specific predictions of the impact of habitat fragmentation: Local extinction of birds in the box-ironbark forests of central Victoria, Australia", Biological conservation, vol. 82, pp. 147-155, 1997.

106. T. Allander, X. Forns, S. U. Emerson, R. H. Purcell and J. Bukh, " Hepatitis C Virus Envelope Protein E2 Binds to CD81 of Tamarins ", Virology, vol. 277, pp. 358-367, 2000.

107. P. Pileri et al., "Binding of hepatitis C virus to CD81", Science, vol. 282, pp. 938-941, 1998.

108. M. Dumontier, K. Michalickova, and C.W.V. Hogue, "Species-specific protein sequence and fold optimizations", BMC Bioinformatics, vol. 3, no. 1, pp. 39, 2002.

109. C. Linnaeus, System of nature, or the three systematic rules of nature proposed by classes, orders, genera, and species. 1735.

110. E. Haeckel, *Generelle Morphologie der Organismen.* Gruyter Press, 1866.

111. H. Copeland "The Kingdoms of organisms". Quarterly review of biology vol.13, pp. 383–420, 1938.

112. R. H. Whittaker, "New concepts of Kingdoms or organisms. Evolutionary relations are better represented by new classifications than by the traditional two Kingdoms". Science, vol. 163, no. 863, 150–60. 1969.

113. C. R. Woese, G. E. Fox, "Phylogenetic structure of the prokaryotic domain: the primary Kingdoms". Proc. Natl. Acad. Science, vol. 74, no. 11, pp. 5088–5090, 1977.

114. C. R. Woese , O. Kandler, M. L. Wheelis, "Towards a natural system of organisms: proposal for the domains Archaea, Bacteria, and Eucarya". Proc.

Natl. Acad. Science, vol. 87, no. 12, pp. 4576–4579. 1990.

115. T. Cavalier-Smith, "A revised six-Kingdom system of life". Biol Rev Camb Philos Soc, vol. 73, no. 3, pp. 203–66, 1998.

116. ftp://ftp.wwpdb.org/pub/pdb/derived_data/pdb_seqres.txt

117. RCSB PDB Web Services, http://www.rcsb.org/robohelp/#webservices/summary.htm

118. W. Kabsch, and C. Sander, "Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features", Biopolymers, vol. 22, no. 12, pp. 2577-2637, 1983.

119. B. Chen, P.C. Tai, R. Harrison, and Y. Pan, "Novel clustering algorithm combined with DSSP post processing for protein sequence motif discovering", Granular Computing, IEEE International Conference on, pp. 449- 452, 2006.

120. T. Wu, J. Mao, and L. Zhang, " An efficient Method for Protein Secondary Structure Prediction", Bioinformatics and Biomedical Engineering, ICBBE 2007, The 1st International Conference on, pp. 21-24, 2007.

121. S.K. Riis, and A. Krogh, "Improving Prediction of Protein Secondary Structure using Structured Neural Networks and Multiple Sequence Alignments", J. Comput. Biol, vol. 3, pp. 163-183, 1996.

122. S.F. Altschul, T.L. Madde, A.A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs" , Nucleic Acids Res., vol. 25, no. 17, pp. 3389–3402, 1997.

123. D.T. Jones, "Protein secondary structure prediction based on position-specific scoring matrices", J. Mol. Biol,, vol. 292, pp. 195-202, 1999.

124. H. Kaur, and G.P.S. Raghava, "Prediction of α-turns in proteins using

PSI-BLAST profiles and secondary structure information", Proteins: Structure, Function, and Bioinformatics, vol. 55, no. 1, pp. 83-90, 2004.

125. B. Rost, "'Review: protein secondary structure prediction continues to rise", J. Struct. Biol., vol. 134, no. 2, pp. 204-218, 2001.

126. G. Pollastri, D. Pryzbylski, B. Rost, and P. Baldi, "Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles", Proteins, vol. 47, pp. 228-235, 2002.

127. ftp://ftp.ncbi.nih.gov/

128. W. Zheng, "Protein secondary structure prediction by combining hidden Markov models and sliding window scores", International Journal of Bioinformatics Research and Applications archive, vol. 1, no. 4, pp. 420-428 , 2005.

129. W. David, and D.W. Mount, *Bioinformatics: sequence and genome analysis*. Cold Spring Harbor Laboratory Press, 2004.

130. J. Garnier, J-F Gibrat, B. Robson, "GOR secondary structure prediction method version IV", Meth Enzymol, vol. 266, pp. 540-553, 1996.

131. A. Clayton,Z. Yanqing, " Neural networks with resilient propagation for protein secondary structure prediction", Granular Computing, IEEE International Conference on, page(s), pp. 766- 769, 2006.

132. G. Wang, R.J. Dunbrack, "PISCES: a protein sequence culling server", Bioinformatics, vol. 19, no. 12, pp. 1589-1591, 2003.

133. The MathWorks, accelerating the pace of engineering and science, " http://www.mathworks.com/ ".

134. P. Gianluca, D. Przybylski, B. Rost, and P. Baldi, "Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles," Proteins: Structure, Function, and

Genetics, vol. 47, no. 2, pp. 228-235, 2002.

135. J. Roman, and A. Jameel, "Backpropagation and recurrent neural networks in financial analysis of multiple stock market returns," Proceedings of the Twenty-Ninth Hawaii International Conference on System Sciences, vol. 2, pp. 454-460, 1996.

136. Jayadeva and S.A. Rahman, "A neural network with O(N) neurons for ranking N numbers in O(1/N) time," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 51, no. 10, 2044-2051, 2004.

137. J. Feng, C.K. Tse, and F.C.M. Lau, "A neural-network-based channel-equalization strategy for chaos-based communication systems," IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 50, no. 7, pp. 954-957, 2003.

138. J. Robert, R. J. Howlett,J. Kacpizyk, *Radial basis function networks 1: recent developments in theory and applications*. Physica Verlag Rudolf Liebing KG Press, 2001.

139. D. F. Specht., *Probabilistic neural networks and general regression neural networks.* McGraw-Hill Press, (1996).

140. M. P. Wachowiak,A. S. Elmaghraby, R. Smolikova, J. M. Zurada, " Generalized regression neural networks for biomedical imageinterpolation", Neural Networks. Proceedings. IJCNN '01. International Joint Conference on, Volume 3: 2133-2138, (2001).

141. W. Gueaieb, "Intelligent Systems Design course", Ottawa University, 2007.

142. J. Heaton, *Introduction to neural networks with Java*. Heaton research Press, 2005.

143. W. Thomas, M. Thomas, P. Werbos, and R. Sutton. *Neural Networks for*

Control. MIT Press, 1995.

144. D. Jesús, and M.T. Hagan, "Backpropagation Through Time for a General Class of Recurrent Network", Proceedings of the International Joint Conference on Neural Networks, Washington, DC, pp. 2638-2642, 2001.

145. D. Jesús, and M.T. Hagan, "Forward Perturbation Algorithm for a General Class of Recurrent Network," Proceedings of the International Joint Conference on Neural Networks, Washington, DC, pp. 2626-2631, 2001.

146. L. Ljung, System identification: Theory for the user. Prentice Hall PTR Press, 1999.

147. J. L. Elman , "Finding structure in time", Cognitive Science, volume 14(2): 179-211, 1990.

148. Q.L. Ji, and W.M. Qi, "The property of PID Elman Neural Network and its application in identification of hydraulic unit", IEEE international conference on control and automation, China, pp. 1795-1798, 2007.

149. S. Ding, W. Jia, C. Su, X. Xu. and L. Zhang, PCA-Based Elman Neural Network Algorithm. Springer Berlin Heidelberg Press, 2008.

150. "Neural Network Toolbox 6.0.2, Design and simulate neural networks", http://www.mathworks.com/

151. D. Nguyen and B. Widrow, "Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights", Proceedings of the international joint conference on neural networks, vol. 3, pp. 21-26, 1990.

152. E. Parzen, "On estimation of a probability density function and mode", Annals of Mathematical Statistics, vol. 33, pp. 1065-1076, 1962.

153. F. Ancona, A.M. Colla, S. Rovetta, and R. Zunino, "Implementing Probabilistic Neural Networks", Neural Comput & Applications, vol. 5, pp.

152-159, 1997.

154. D.F. Specht, "Probabilistic neural networks", Neural Networks, vol. 3, no. 1, pp. 109–118, 1990.

155. Probabilistic Neural Networks , http://www.mathworks.com/

156. R.O. Duda, P.E. Hart and D.G. Stork, *Pattern Classification*. Wiley-Interscience Press, 2000.

157. D.A. Singer, and R. Kouda, "Implementing probabilistic Neural Networks", Neural Computing & Applications, Springer London, vol. 5, no. 3, pp. 152-159, 2005.

158. I. Galleske, and J. Castellanos, "Optimization of the Kernel Functions in a Probabilistic Neural Network Analyzing the Local Pattern Distribution", Neural Computation, vol. 14, pp. 1183–1194, 2002.

159. A. Sawhney, and A. Mund, "Adaptive Probabilistic Neural Network-based Crane Type Selection System", J. Constr. Engrg. and Mgmt., vol. 128, no. 3, pp. 265-273, 2002.

160. J. R. Green, J. Michael,and M. J. Korenberg, "Nonlinear System Identification Provides Insight Into Protein Folding", Electrical and Computer Engineering, CCECE '06. Canadian Conference on, pp. 721-724, 2006.

161. B. Bolat, and T. Yıldırım, Active Learning for Probabilistic Neural Networks. Springer Berlin Press, 2005.

162. M. Reczko, " Protein secondary structure prediction with partially recurrent neural networks ", SAR QSAR Environ Res., vol. 1, no. 2-3, pp. 153-159, 1993.

163. R.D. Reed, and R.J. Marks, Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks. MIT Press, 1999.

164. R. Guerin, H. Ahmadi, and M. Nagshienh, "Equivalent capacity applications to bandwidth allocations in high speed networks", IEEE journal on selected areas in communication, vol. 9, pp. 968-998, 1991.

165. K. P. O'Brien, M. Remm, and E. L. L. Sonnhammer, " Inparanoid: a comprehensive database of eukaryotic orthologs", Nucleic Acids Res, vol. 33, D476.D480, 2005.

166. M. Remm, C.E. Storm, and E.L. Sonnhammer, "Automatic clustering of orthologs and in-paralogs from pairwise species comparisons", *J. Mol. Biol.*, vol. 314, pp. 1041–1052, 2001.

167. K.P.O'Brien, I. Westerlund, and E.L. Sonnhammer, "OrthoDisease: a database of human disease orthologs" , *Hum. Mutat.*, vol. 24, pp. 112–119, 2004.

168. E.L. Sonnhammer and E.V. Koonin, "Orthology, paralogy and proposed classification for paralog subtypes" , *Trends Genet.*, vol.18, pp. 619–620, 2002.

169. "InParanoid: Eukaryotic Ortholog Groups 35 organisms: 610047 sequences", http://inparanoid.sbc.su.se/cgi-bin/index.cgi.

170. R. Tatusov, M. Galperin, D. Natale, and E. Koonin, "The COG database: a tool for genome-scale analysis of protein functions and evolution", Nucleic Acids Res, vol. 28, pp. 33-36, 2000.

171. R. Tatusov, N. Fedorova, J. Jackson, A. Jacobs, B. Kiryutin, E. Koonin, D. Krylov, R. Mazumder, S. Mekhedov, A. Nikolskaya, B. Rao, S. Smirnov, A. Sverdlov, S. Vasudevan, Y. Wolf, J. Yin, and D. Natale, "The COG database: an updated version includes eukaryotes" , BMC Bioinformatics, vol. 4, pp. 41, 2003.

172. V. Noort, B. Snel, M. Huynen, "Predicting gene function by conserved co-

expression", Trends Genet, vol. 19, pp. 238-242, 2003.

173. L. Li, C. Stoeckert, D. Roos ," OrthoMCL: identification of ortholog groups for eukaryotic genomes", Genome Res, vol. 13, pp. 2178-2189, 2003.

174. H. Tim, A. H. Martijn, D. V. Jacob, and M. G. Peter, "Benchmarking ortholog identification methods using functional genomics data", Genome Biol, vol. 7, no. 4, 2006.

175. M. G. Conte, S. Gaillard, G. Droc, and C. Perin, "Phylogenomics of plant genomes: a methodology for genome-wide searches for orthologs in plants ", BMC Genomics. vol. 9, no. 183, 2008.

176. J. C. Chiu, E. K. Lee, M. G. Egan, I. N. Sarkar, G. M. Coruzzi, and R. DeSalle, "OrthologID: automation of genome-scale ortholog identification within a parsimony framework ", Bioinformatics , vol. 22, no. 6, pp. 699-707, 2006.

177. InParanoid: Eukaryotic Ortholog Groups, " http://inparanoid.sbc.su.se/".

178. R.L.Plackett, "Karl Pearson and the Chi-Squared Test". International Statistical Review, vol. 51, no. 1, pp. 59–72, 1983.