Protein Secondary Structure Prediction Using Artificial Neural Networks

Dilip Antony Joseph
CS00294, Department of Computer Science and Engineering
Indian Institute of Technology, Madras

September 10, 2002

Contents

1 Introduction **4**
1.1 Structure of Proteins 4
1.2 Why Secondary Structure(SS) Prediction? 4
1.3 Methods of Secondary Structure Prediction 6

2 Artificial Neural Networks **6**
2.1 An Artificial Neuron 7
2.2 Feed Forward Networks 7
2.3 Training The Network 8

3 Input and Output Encoding **8**
3.1 Output Representation 9
3.2 Input Representation 9
3.2.1 What does PSIBLAST do? 9
3.2.2 Windowing the Sequence 10
3.2.3 Encoding at the Two ends of the Sequence 10

4 Measuring Prediction Accuracy **11**
4.1 Q3 Score 11
4.2 CASP Test Sets 11

5 Training Data **12**
5.1 Choosing the Sequences - PDB SELECT 12
5.2 FASTA Sequence and Secondary Structure Information 12
5.3 PSIBLASTing 13
5.4 Balanced Training Set 13

6 Prediction Using Feed Forward Neural Networks **13**
6.1 Network Architecture 13
6.2 Network Parameters 14
6.3 Implementation 15
6.4 Limitations 16

7 Jury Predictors **16**

8 Results and Observations **16**
8.1 Prediction using Feed Forward Neural Networks 16
8.1.1 Unbalanced Training Set 17
8.1.2 Training on 1740 sequences 17
8.1.3 Training on 6529 sequences 20
8.1.4 Effect of Window Size 20
8.1.5 Effect of Number of Training Epochs 21
8.1.6 Effect of Learning Rate and Momentum Factor 23
8.2 Jury of Networks 24

9 Conclusion **24**

Abstract

Proteins are the machinery of life. Deciphering the structure of proteins can help in the design of new drugs and medicines. The various experimental methods to determine the complex 3D structure of proteins are often very time-consuming and sometimes, even impossible. Therefore computers are used to predict the Secondary Structure of proteins. The Secondary Structure thus predicted can give insights into the 3D structure. Of the various methods used in Secondary Structure Prediction, Artificial Neural Network based predictors have proven to be the most effective. In this paper, the focus is on the implementation of a Feed Forward Neural Network based Protein Secondary Structure predictor. A Jury of Networks is also used to improve the prediction efficacy. The various parameters affecting the network training and prediction are analyzed.

1 Introduction

Proteins are the machinery of life. They are involved in all bodily functions - be it catalysis of reactions, transport of nutrients or transmission of signals to various parts of the body. Understanding the structure and specific functions of proteins can lead to the design of new drugs which can effectively combat the diseases plagueing humanity.

It is known that the structure of the protein determines its function. In aqueous environments, the proteins fold up into unique complex 3D structures. So the onus over the years has been on deciphering the structure of proteins, and thus to determine their biological functions.

1.1 Structure of Proteins

Proteins are long polypeptide chains made up of 20 different amino acids. The sequence of amino acids occurring in the protein determines the *Primary Structure* (Figure 1). The *Tertiary* (Figure 3) or 3-dimensional structure is determined by the complex folding process that takes places in aqueous media. This tertiary structure is the factor which determines the protein function. Intermediate to the primary and tertiary structures, the *Secondary Structure* (Figure 2) classifies each amino acid in the amino acid sequence as - *Alpha Helix (H)* or *Beta Strand (E)* or *Coil (C)*. The Secondary Structure gives insights into the manner in which the protein folds to form the unique 3D structure.

1.2 Why Secondary Structure(SS) Prediction?

Knowledge of the 3D structure of proteins is a very important factor in medical research. The experimental methods (eg. crystallography) used to determine the protein tertiary structure are very time consuming and sometimes impossible. It is easier to first determine the Secondary Structure from the amino acid sequence. Using the secondary structure thus determined, we can predict the tertiary structure through methods like threading. The tertiary structure can give insights into the functions of the protein.

As a result of large scale genome sequencing projects, the sequence-structure gap is rapidly increasing. The number of known amino acid sequences greatly exceeds the number of known structures. Thus there is a

VVSLAGRDLLCLQDYTAEEIVTILETAKMFKIVGKIGKPHRLLEGKTLAMIFQKPSTRTRVSFEVAMAHLGGHALYLNAQDLQLR
RGETIADTARVLSRYVDAIMARVYDHKDVEDLAKYATVPVINGLSDFSHPCQALADYMTIWEKKGTIKGVKVVYVGDGNNVAH
SLMIAGTKLGADVVVATPEGYEPDEKVIKWAEQNAAESGGSFELLHDPVKAVKDADVIYTDVWASMGQEAEAEERRKIFRPFQ
VNKDLVKHAKPDYMFMHCLPAHRGEEVTDDVIDSPNSVVWDQAENRLHAQKAVLALVMGGIKF

Figure 1: Primary Structure. A Sequence of the 20-letter Amino Acid Alphabet

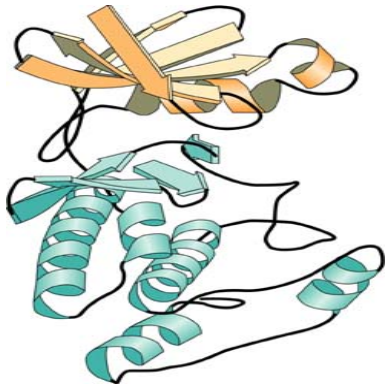

Figure 2: Alpha Helices and Beta Strands. Alpha Helices are shown in Green and Beta Strands in Orange.

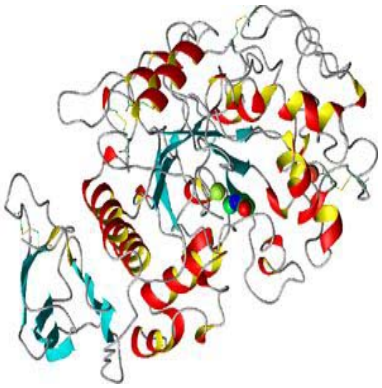

Figure 3: Tertiary Structure of Amylase. The 3D folding of various alpha helices, beta sheets and coils can be seen

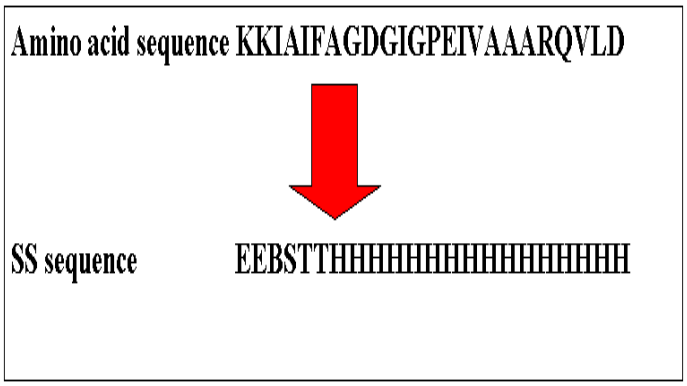

Figure 4: The Neural Network maps the Amino Acid String to the corresponding Secondary Structure String.

need for computational methods that can predict the protein structure, in a reasonable amount of time.

1.3 Methods of Secondary Structure Prediction

Existing methods for Secondary Structure prediction from amino acid sequence involve the use of Hidden Markov Models (HMMs) or Artificial Neural Networks (ANNs). Neural Networks that leverage the evolutionary information are currently on the top in terms of prediction accuracy. In this paper, the focus is on the use of Feed Forward Neural Networks in Secondary Structure prediction.

2 Artificial Neural Networks

Artificial Neural Networks are *computational models* which have the ability to adapt or learn, to generalize, or to cluster or organise data. They attempt to model the functioning of the brain. In protein Secondary Structure prediction, the neural network learns to predict the correct Secondary Structure string given the Amino acid string (Figure 4). The basic unit of an Artificial Neural Network is a *neuron*. These neurons interact with other neurons through weighted connections. Using an ANN for prediction involves three phases:

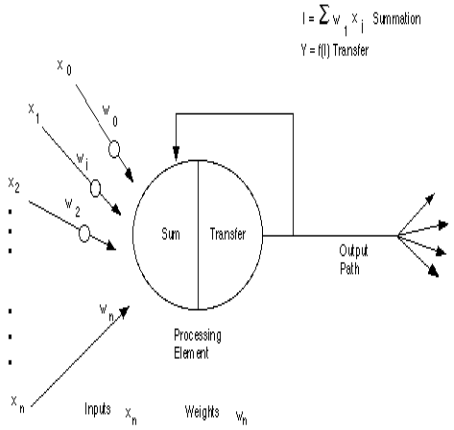

Figure 5: A Single Neuron

- Training
- Testing
- Prediction

2.1 An Artificial Neuron

The structure of a single neuron is shown in Figure 5.
The neuron performs a weighted sum over all its inputs. The output of the neuron is calculated by applying an activation function on this weighted sum. This output is then passed on as input to the other neurons. The most commonly used activation function is the *Sigmoid function*

$$f(x) = \frac{1}{1+e^{-x}}$$

2.2 Feed Forward Networks

Feed Forward Neural Networks consist of neurons arranged into distinct layers. There are no connections between any two neurons in the same layer. The network consists of an *input layer*, an *output layer* and a variable number of *hidden layers* (Figure 6). The input, applied at the input layer, is

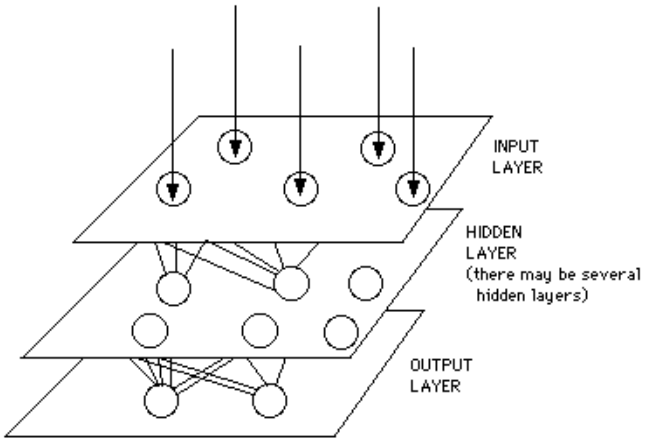

Figure 6: A Feed Forward Neural Network with a single Hidden Layer

propagated in the forward direction, through the hidden layers, uptill the final output layer. The output of a neuron is not fed back into any neuron in any previous layer, at any stage.

2.3 Training The Network

Training a Neural Network consists of presenting to the network, a set of known input-output pairs. The weights of the network are adjusted so as to minimize the error between the desired output and the output calculated by the network. An *epoch* consists of presenting all the training pairs in the set to the network a single time. Training consists of many epochs, with the order of training pairs randomly jumbled at each epoch. The network weights adjustment is carried out using the popular *Back Propagation rule* for Feed Forward Networks.

3 Input and Output Encoding

The representation of the input and output of the Neural Network is an important factor to be considered during the network design. The Neural Network is trained to recognise and adapt to patterns in the input. So the representation of the input plays an important role in determining the efficacy

of the network.

3.1 Output Representation

The network classifies the current residue to be predicted into 3 distinct classes:

- Alpha Helix - H
- Beta Strand - E
- Coil or Loop - C

Orthogonal Encoding is used to represent the 3 output states: H - 100, E - 010, C - 001

3.2 Input Representation

Orthogonal encoding for the input needs 20 numbers per residue. For example, Alanine(A) is 10000 00000 00000 00000, Leucine(L) is 01000 00000 00000 00000, etc. Orthogonal encoding ensures that the distance between the encodings of any two residues is identical. Therefore, the order of assigning codes to the 20 amino acids is irrelevant.

It has been observed that using evolutionary information in the prediction process leads to higher prediction accuracies. In the predictor implemented in this project, the input consisted of *PSIBLAST[13] [14] profiles* and not the orthogonal encoding of the amino acid sequence.

3.2.1 What does PSIBLAST do?

PSIBLAST stands for *Position Specific Iterative* Basic *Local Alignment Search Tool*. It is a freely available tool to perform a multiple alignment of a given sequence against a specified protein database. PSIBLAST produces a profile matrix which gives the frequency of each of the 20 amino acids in each position of the sequence. Therefore, for each amino acid position, PSIBLAST produces 20 numbers usually in the range -7 to 7. These values after being scaled down to the range 0 to 1 using the sigmoid function, are used as input to the neural network. Evolutionary information comes into play here as a result of the multiple alignment.

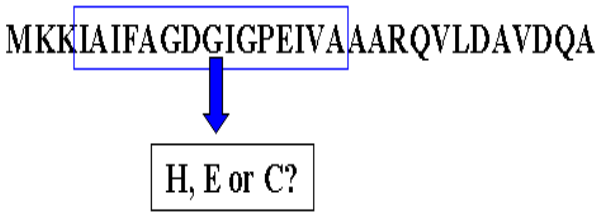

Figure 7: The Neural Network predicts the state of the central residue of the window. This window is slid over the entire sequence.

3.2.2 Windowing the Sequence

The state of an amino acid (H, E or C) depends not only on the nature of the residue, but also on the surrounding residues. So the network is fed with a *window* of residues. In most of the predictors developed in this project, a window size of *15* was used. When fed with this window of 15 amino acids, the network predicts the Secondary Structure state of the central residue of the window(Figure 7).

It is known that, in most cases, the state of the amino acid is also influenced by amino acids far away from it in the sequence. A window of 15 will not be able to capture these long range relationships. Increasing the window size is not a solution. Using larger window sizes often leads to *overfitting*.

3.2.3 Encoding at the Two ends of the Sequence

A suitable encoding scheme has to be designed to handle the cases when the window of residues spans outside the sequence. This happens for the first and last seven residues (assuming window size of 15). To handle these cases, the number of values needed to represent a sequence position was increased to 21. For actual amino acid positions, the input consisted of the 20 numbers obtained from the PSIBLAST profile. The 21^{st} unit was taken to be 0.0. The positions outside the amino acid sequence were encoded as 00000 00000 00000 00000 1 (20 0s followed by a 1).

4 Measuring Prediction Accuracy

4.1 Q3 Score

The most common measure of the prediction accuracy is the *Q3 score*. The Q3 score is calculated as follows:

$$\text{Q3} = \frac{N_a + N_b + N_c}{N_t} * 100$$

where

N_a = Number of residues correctly predicted in the Helix State
N_b = Number of residues correctly predicted in the Strand State
N_c = Number of residues correctly predicted in the Coil State
N_t = Total number of residues in the sequence

Prediction accuracy is judged on the basis of the average Q3 score of a set of test sequences.

The individual percentages of the alpha helices, beta strands and coils predicted correctly is also noted. Coils are the easiest to predict, while beta strands with the their long range interactions are the most difficult. The individual percentages of each residue are monitored to ensure that a good percentage of all three types of residues are predicted correctly.

4.2 CASP Test Sets

CASP stands for Critical Assessment of Techniques for Protein Structure Prediction. It is a community wide experiment held every two years, aimed at establishing the current state of the art in protein structure prediction, identifying what progress has been made, and highlighting where future effort may be most productively focussed [15].

The protein sequences used in the CASP3 (1998) and CASP4(2000) prediction contests were used to test the efficacy of the predictor. 34 and 33 sequences of the CASP3 and CASP4 prediction sets respectively, for which the correct secondary structure details were available, were used to test the predictor.

5 Training Data

The neural networks have to be trained with existing data before they can be used for predicting the secondary structure of new protein sequences. The following data were needed for each training sequence considered.

- Sequence information in FASTA format
- Secondary Structure String (H,E and C classes)
- PSIBLAST[13] Profile

5.1 Choosing the Sequences - PDB SELECT

It is important that the network be trained on a comprehensive set of sequences, covering all possible kinds of folds. To increase the effectiveness of the training, the train set constructed should not contain sequences which are very similar to each other. Keeping in mind this factor, the training set used here was picked from the *PDB SELECT* Database [11][12]. The PDB-SELECT database is a subset of the structures in the PDB[10] that does not contain highly homologous sequences. The database contains two lists of protein sequences which listed protein sequences with mutual sequence similarity less than 25% and 90% respectively. Subsets of these two lists were used to train the network. The results and observations are discussed in Section 8.

5.2 FASTA Sequence and Secondary Structure Information

The PDB ids of the chosen sequences were parsed from the PDB SELECT database. The FASTA and Secondary Structure information corresponding to these ids were automatically retrieved from the PDB at *http://www.rcsb.org*. The Secondary Structure was parsed out directly from the HTML file which contained the sequence details. Separate PERL scripts and C programs were written to perform the above tasks.

5.3 PSIBLASTing

The networks used for prediction were fed with the PSIBLAST profiles of the sequences. It was thus necessary to obtain the PSIBLAST profiles of the sequences under consideration before the network training was started. This was done with the help of the stand-alone version of PSIBLAST obtained from *http://www.ncbi.nlm.nih.gov/BLAST/*. A custom PERL script handled the automatic PSIBLASTing of all the selected sequences, as well as the parsing of the generated log files to obtain the required profiles. The profiles obtained were stored in files named `PDBID.mt`.

5.4 Balanced Training Set

Balancing the training set in terms of the number of residues present in each of the three states affects the efficacy of the predictor. Naturally occurring proteins contain a large proportion of Coil states. As a result, it is easiest to predict the coil states. Helices come next in terms of ease of prediction. If the sequences are not carefully chosen, it is highly possible that the proportion of beta strands and alpha helices in the training set are very low when compared to that of the coils. In such a case, the network will not be able to identify helices and strands. This undesired behaviour observed in the predictor is described in Section 8.

6 Prediction Using Feed Forward Neural Networks

A Feed Forward Neural Network based Secondary Structure Predictor was implemented in the first phase of this project. The details of the predictor are as follows.

6.1 Network Architecture

The predictor consisted of two different Neural Networks. Both networks consisted of 3 layers - input layer, hidden layer and output layer.

The first network was fed with fixed size windows of PSIBLAST profiles of the input amino acid sequence. The output layer of this network used orthogonal encoding to represent the three different classes - H, E and C.

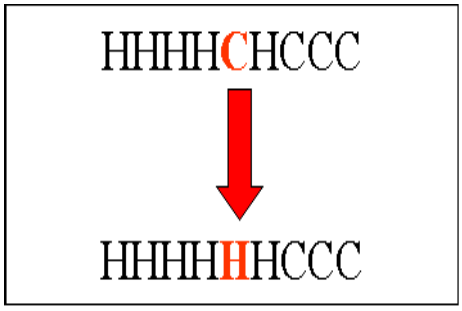

Figure 8: 'Cleaning up' – It is most likely that the residue marked coil(C) in the middle of sequence of Alpha Helices(H) has been wrongly predicted. The second level network outputs the corrected Secondary Structure Sequence.

The state of the central amino acid in the window formed the output of the network. The states of all the amino acids in the sequence are predicted by the first network by sliding the window along the length of the sequence. This output of the first network, i.e. a string of H, E and Cs corresponding to the predicted secondary structure are fed into the second neural network. The function of the second network is to 'clean up' the output of the first network (Figure 8). It has been observed that the length of the helix and strand segments predicted by single network predictors are often very small compared to the helix and strand segments observed in nature. Double Neural Network predictors aim to increase the efficacy with respect to the length distribution of the coils and strands.

6.2 Network Parameters

Squashing function

All the neurons in both networks used the following squashing function:

$$f(x) = \frac{1}{1+e^{-x}}$$

Layers

Both networks used in the predictor consisted of 3 layers. The details of the various layers are shown in Table 1. The *Learning Rate* and *Momentum Factor* were varied over different training runs and the performance of the network was studied for each case.

Network 1	
Number of Layers	3
Number of Neurons in Input Layer	315
Number of Neurons in Hidden Layer	75
Number of Neurons in Output Layer	3
Network 2	
Number of Layers	3
Number of Neurons in Input Layer	60
Number of Neurons in Hidden Layer	60
Number of Neurons in Output Layer	3

Table 1: Layer details of the Protein Secondary Structure Predictor based on double Feed Forward Neural Networks

Weights

The weights of the two networks used in the predictor were initialized with random values between -0.005 and +0.005. This range was intuitively chosen to avoid the saturation of the neuron outputs to 0 or 1, after applying the logistic function.

6.3 Implementation

The Feed Forward Neural Network based predictor program was written in *C++*. An Object oriented approach was followed in the design of the program. A Layer of neurons was modelled as a class. The Neural Network class contained an array of Layer objects, along with functions like FeedForward and Backpropagate. The Predictor class prepares the Protein Sequence input in the right format and feeds it to the neural network objects. Specialized predictors can be further inherited from the generic Predictor class.

The objecti-oriented approach proved to be very helpful in debugging the code. The Neural Network class was thoroughly tested and debugged before being used in the Predictor class. Thus for the Predictor class, the Neural Networks were just 2 blackboxes. This modular approach also makes it very easy to extend the Neural Network. For example, adding a new layer to the network takes only 2 additional lines.

6.4 Limitations

Feed Forward Neural Networks use a fixed window (say, of size 15) of amino acids to predict the state of the central residue. However, it is well known that the secondary structure state of an amino acid is influenced even by amino acids far away in the sequence. The complex 3D folding of proteins can result in sequence-wise distant amino acids occurring very close to each other in the tertiary structure. These long distance relationships are very easily observed in beta strands. Therefore, a window of size 15 cannot capture the complete information needed to predict the secondary structure state of the central amino acid. Increasing the window size leads to overfitting, which is discussed in Section 8.

7 Jury Predictors

A jury of networks based predictor consists of many independently trained networks. The assignment of various secondary structure states to the amino acid residues is based on the majority of vote among the independent predictors.

The predictors forming the jury may be of various types and sizes. There can be specialized predictors in the jury, which can utilize biological information to predict a particular state (say alpha helices or beta strands only) more accurately . The simple majority vote decision may be replaced by a custom weighting scheme which takes into account the observed accuracies of the individual predictors constituting the jury.

In the jury based predictor developed in this project, only Feed Forward Neural networks trained on different training sets and at different learning rates were used. The scores obtained by this predictor is discussed in Section 8.2.

8 Results and Observations

8.1 Prediction using Feed Forward Neural Networks

The predictor was trained on three different subsets of the PDB SELECT database. Various parameters of the predictor - learning rate, momentum

factor, window size - were varied. The results obtained are summarized in this section.

8.1.1 Unbalanced Training Set

The first training set chosen consisted of 30 sequences randomly chosen from the PDB SELECT 25 database (sequence similarity $< 25\%$). The Q3 scores obtained on testing the network on the CASP 3 and CASP 4 sequences were below 60%. The coil states predicted correctly accounted for the bulk of the Q3 score. Not a single beta strand was predicted! This was due to the very low proportion of beta strands in the training set.

A second training set of 60 sequences was constructed by adding sequences with a large number of residues in the beta strand state. The so-trained network gave similar Q3 scores, but alpha helices were not predicted at all! In this case, the balance was tilted by the presence of a large number of beta strands.

The third training set of 115 sequences was chosen such that the proportions of residues in the three states - H, E or C, were approximately equal. This network gave better results with the CASP 3 and CASP 4 test sets. The results are summarised in 2. We can see that the second 'cleanup' neural network improved the Q3 score only marginally. But the marginal increase was accompanied by an increase in the percentage of residues correctly predicted to be in the beta strand state, usually the most difficult state to predict.

8.1.2 Training on 1740 sequences

A training set with 1740 sequences was constructed from the PDB SELECT 25% list by removing the sequences which were present in the CASP 3 and CASP 4 test sets. Each training epoch on this training set took approximately 15 minutes, leading to a 2 day training period for 200 epochs. An IBM PC with Pentium 4 1.4 GHz processor and 512 MB of RAM was used in all training runs. The scores of this predictor on the CASP 3 and CASP 4 test sets are shown in Table 3. The Q3 scores have increased significantly from around 66% to 75%. This is because the network encountered a wider variety of sequences and their distinct secondary structures in its training phase.

Number of Training Sequences : **115**			
Window Size : **15**		Total Number of Residues : **16204**	
Learning Rate : **0.005**		Momentum factor : **0.9**	
CASP 3			
After Network 1		**After Network 2**	
average Q3 Score	*67.47*	average Q3 Score	*68.19*
% of correctly predicted residues in each state			
Alpha Helix	*74.77%*	Alpha Helix	*76.19%*
Beta Strand	*54.91%*	Beta Strand	*56.46%*
Coil	*66.60%*	Coil	*66.25%*
CASP 4			
After Network 1		**After Network 2**	
average Q3 Score	*67.47*	average Q3 Score	*68.25*
% of correctly predicted residues in each state			
Alpha Helix	*77.49%*	Alpha Helix	*78.04%*
Beta Strand	*60.19%*	Beta Strand	*61.06%*
Coil	*66.20%*	Coil	*66.46%*

Table 2: Performance of the Predictor trained on a balanced training set consisting of 115 sequences taken from PDB SELECT.

Number of Training Sequences : **1740**			
Window Size : **15**		Total Number of Residues : **349083**	
Learning Rate : **0.005**		Momentum factor : **0.9**	
CASP 3			
After Network 1		**After Network 2**	
average Q3 Score	*76.79*	average Q3 Score	*77.27*
% of correctly predicted residues in each state			
Alpha Helix	*80.06%*	Alpha Helix	*78.24%*
Beta Strand	*66.69%*	Beta Strand	*68.04%*
Coil	*78.50%*	Coil	*80.09%*
CASP 4			
After Network 1		**After Network 2**	
average Q3 Score	*74.85*	average Q3 Score	*76.17*
% of correctly predicted residues in each state			
Alpha Helix	*82.62%*	Alpha Helix	*82.79%*
Beta Strand	*59.54%*	Beta Strand	*61.18%*
Coil	*78.25%*	Coil	*80.59%*

Table 3: Performance of the Predictor after training on 1740 sequences from the PDB SELECT 25% list. The CASP 3 and CASP 4 sequences found in the PDB SELECT were removed.

Number of Training Sequences : **6529**			
Window Size : **15**		Total Number of Residues : **1436264**	
Learning Rate : **0.005**		Momentum factor : **0.9**	
CASP 3			
After Network 1		**After Network 2**	
average Q3 Score	*76.79*	average Q3 Score	*77.27*
% of correctly predicted residues in each state			
Alpha Helix	*80.06%*	Alpha Helix	*78.24%*
Beta Strand	*66.69%*	Beta Strand	*68.04%*
Coil	*78.50%*	Coil	*80.09%*
CASP 4			
After Network 1		**After Network 2**	
average Q3 Score	*74.85*	average Q3 Score	*76.17*
% of correctly predicted residues in each state			
Alpha Helix	*82.62%*	Alpha Helix	*82.79%*
Beta Strand	*59.54%*	Beta Strand	*61.18%*
Coil	*78.25%*	Coil	*80.59%*

Table 4: Performance of the Predictor after training on 6529 sequences from the PDB SELECT 90% list. The CASP 3 and CASP 4 sequences found in the PDB SELECT were removed.

8.1.3 Training on 6529 sequences

The next training set used consisted of 6529 sequences from the PDB SELECT 90 % list, with the sequences present in CASP 3 and CASP 4 removed. The presence of sequences that are upto 90 % similar to each other, undermines the effectiveness of this training set. Each epoch on this training set took 50 minutes. Training for 300 epochs was completed in a period of 10 days. The results obtained are shown in Table 4.

8.1.4 Effect of Window Size

The window size chosen affects the efficacy of the predictor to a great extent. It was observed that a window of 13 was unable to capture much of the contextual information needed to predict the secondary structure state of the central residue of the window. It gave Q3 scores of 64.64 and 64.98 for CASP

Number of Training Sequences : **1740**			
Window Size : **13**		Total Number of Residues : **349083**	
Learning Rate : **0.005**		Momentum factor : **0.9**	
CASP 3			
After Network 1		**After Network 2**	
average Q3 Score	*70.56*	average Q3 Score	*64.64*
% of correctly predicted residues in each state			
Alpha Helix	*74.24%*	Alpha Helix	*68.98%*
Beta Strand	*52.29%*	Beta Strand	*43.54%*
Coil	*76.75%*	Coil	*72.00%*
CASP 4			
After Network 1		**After Network 2**	
average Q3 Score	*70.18*	average Q3 Score	*64.93*
% of correctly predicted residues in each state			
Alpha Helix	*75.55%*	Alpha Helix	*71.79%*
Beta Strand	*54.92%*	Beta Strand	*44.20%*
Coil	*74.14%*	Coil	*69.40%*

Table 5: Performance of the Predictor after training with a window size of 13 on 1740 sequences from the PDB SELECT 25% list. The CASP 3 and CASP 4 sequences found in the PDB SELECT were removed. The efficacy of the network was found to be much lower than the networks trained with a window size of 15.

3 and CASP 4 respectively . Using a window size of 17 lead to overfitting of the training data. The CASP 3 and CASP 4 test sets registered Q3 scores of 69.30 and 69.80 respectively. However, the predictor gave 100% results when predicting the sequences in the training set. Therefore, a window size of 15 was chosen to avoid the overfitting problem. Tables 5 and 6 summarize the scores obtained for training runs with window sizes of 13 and 17 respectively. Figure 9 shows the performance of the network when using window sizes of 13, 15 and 17.

8.1.5 Effect of Number of Training Epochs

The number of training epochs directly affects the performance of the network. The network was seen to adapt to the training data after the very

Number of Training Sequences : **1740**			
Window Size : **17**		Total Number of Residues : **349083**	
Learning Rate : **0.005**		Momentum factor : **0.9**	
CASP 3			
After Network 1		**After Network 2**	
average Q3 Score	*69.30*	average Q3 Score	*64.32*
% of correctly predicted residues in each state			
Alpha Helix	*67.67%*	Alpha Helix	*62.95%*
Beta Strand	*56.66%*	Beta Strand	*49.06%*
Coil	*75.93%*	Coil	*71.70%*
CASP 4			
After Network 1		**After Network 2**	
average Q3 Score	*69.80*	average Q3 Score	*65.16*
% of correctly predicted residues in each state			
Alpha Helix	*73.39%*	Alpha Helix	*70.87%*
Beta Strand	*58.31%*	Beta Strand	*47.83%*
Coil	*65.16%*	Coil	*69.74%*

Table 6: Performance of the Predictor after training with a window size of 17 on 1740 sequences from the PDB SELECT 25% list. The CASP 3 and CASP 4 sequences found in the PDB SELECT were removed. The efficacy of the network was found to be much lower than the networks trained with a window size of 15. However the network predicted the sequences in the training set with 100 % accuracy. This can be attributed to overfitting.

first training epoch. The CASP 3 and CASP 4 Q3 scores were fixed as the performance measure of the predictor. The performance of the network gradually increased with the number of training epochs. However after a certain number of epochs, the performance was seen to saturate. It is not useful to train the network beyond this saturation point.

In the training runs carried out in this project, the networks were dumped to disk files at user specified epoch intervals. The network files were used to study the variation of the network performance with the number of training epochs. The results of this experiment can be seen in Figures 9.

Only the networks(dumped on the disk) which gave the maximum Q3 score (corresponding to the highest point in the plots) were further used in the predictor.

8.1.6 Effect of Learning Rate and Momentum Factor

The Learning Rate and Momentum Factor also play a part in determining the effectiveness of the training. The learning rate and momentum factor affect the training process through the following weight correction equation:

$$\delta W_{ij}(t) = L * g_i * y_j + M * \delta W_{ij}(t-1)$$

where

- L = Learning Rate
- M = Momentum Factor
- $\delta W_{ij}(t)$ = Correction to be applied to the connection weight from neuron j to neuron i at time t
- g_i = Error Gradient at neuron i
- y_j = Input to neuron i from the output of neruon j
- t = Time

A network trained at a very low learning rate will take a very long time to converge to its minimum (possibly local) error point. A very high learning rate may cause the network to fluctuate around the minimum position; sometimes never converging. A learning rate of 0.005 was found to be give

Number of Predictors in Jury : **9**	
CASP 3	
Q3 Score	*78.05*
% of Alpha Helices predicted correctly	*74.77%*
% of Beta Strands predicted correctly	*66.89%*
% of Coils predicted correctly	*87.21%*
CASP 4	
Q3 Score	*77.49*
% of Alpha Helices predicted correctly	*74.39%*
% of Beta Strands predicted correctly	*69.56%*
% of Coils predicted correctly	*86.47%*

Table 7: Performance of a Jury of Predictors on the CASP 3 and CASP 4 Test Sets

good results in the training runs conducted in this project. The performance of the network on learning rates of 0.0001, 0.005 and 1.0 are shown in Figure 10.

8.2 Jury of Networks

A Jury of Networks based predictor was observed to give better results during the predictions. On testing with the CASP 3 and CASP 4 sets, the jury gave a Q3 score of 78.05 and 77.49 respectively. The results are summarized in Table 7. Figures 11 and 12 show the amino acid sequence and the correspondence between the predicted structure and the actual structure for 2 different proteins.

9 Conclusion

A Feed Forward Neural Network based Protein Secondary Structure predictor was implemented as part of the project. The efficacy of the network was adjudged on the basis of the Q3 scores obtained for the CASP 3 and CASP 4 test sets. Various parameters of the predictor - window size, learning rate and momentum factor were varied and the predictor performance was measured. A jury of networks based predictor was constructed from

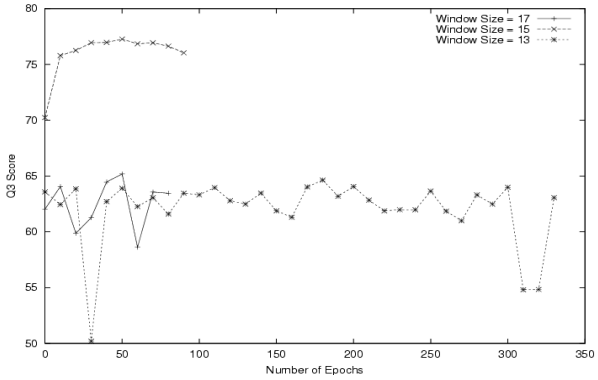

Figure 9: Variation of Predictor performance on the CASP 3 test set with number of training Epochs. The plots for the three window sizes – 13, 15 and 17 are shown here.

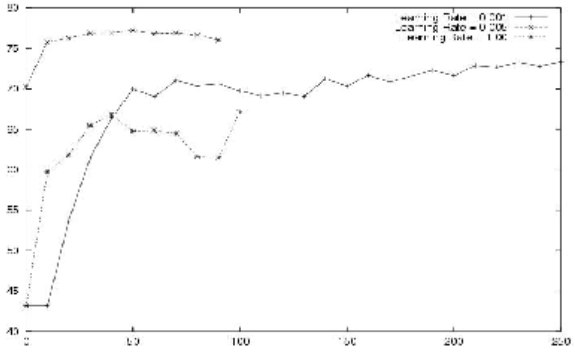

Figure 10: Variation of Network with learning rates of 0.0001, 0.005 and 1.0. A window size of 15 was used in all the three cases. The Q3 scores shown above correspond to the CASP 3 test set. The network trained at a learning rate of 0.005 attains a Q3 score of 77.27 in just after 50 training epochs. At a rate of 0.0001, the network learns very slowly and reaches a Q3 score of 73.3 after 250 epochs. The network trained at a very high learning rate of 1.0 gave the highest Q3 score of 67.12 after 100 epochs. The difference in the time spans of the three graphs is due to the sudden stoppage of the network, due to time constraints.

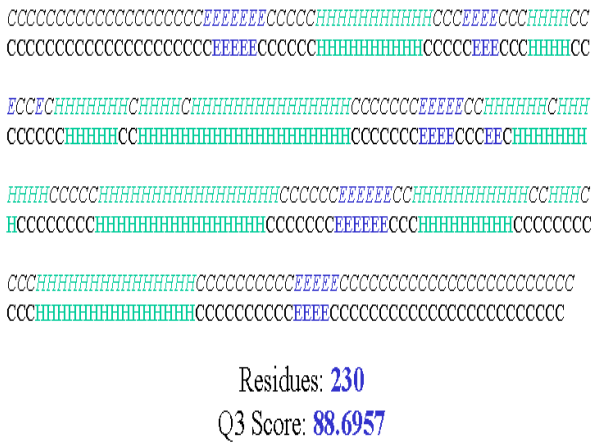

Figure 11: A Good Prediction. The first line in each pair of lines gives the actual Secondary Structure String. The second line displays the predicted structure. The Q3 score of this sequence is 88.69.

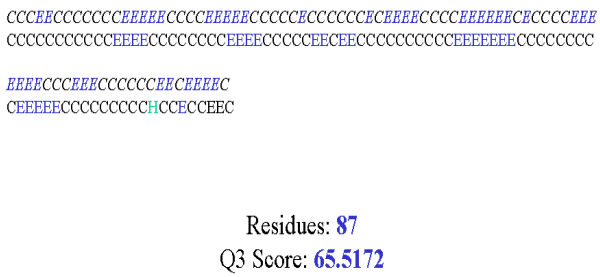

Figure 12: A Bad Prediction. The Q3 score of this sequence is 65.52. Many regions of mismatch between the correct secondary structure and the predicted one can be observed.

independently trained networks .

The jury of Feed Forward Network based predictors developed in this project obtained accuracies very close to that of the best accuracies obtained till date. The performance of the predictor in comparison to the best methods currently existing will be available in December 2002 after the conclusion of the CASP 5 experiment, for which it is a participant. The prediction accuracy showed a very high increase when the training set was expanded from 115 sequences to over 1700 sequences. This accuracy can be further improved by periodically training the network with newly discovered protein structures. Inclusion of a protein fold, radically different from the existing training sequences, into the training set will improve the accuracy of the predictor on similar proteins discovered in the future. Thus the jury based predictor developed here can be used to accurately predict the secondary structure of newly sequenced proteins. This information is very useful in techniques for deciphering the 3D structure of proteins, which is the ultimate aim.

Further work needs to be done to improve the prediction accuracies of Feed Forward Network based juries. The juries can be enhanced by adding predictors (neural network based or others) which use biological knowledge about specific folds and structures to a great extent. Bidirectional Recurrent Neural Networks attempt to overcome the fixed window size of Feed Forward networks by rolling over the sequence from both ends. A jury consisting of all these various predictors, weighted in a appropriate manner, can lead to higher prediction accuracies.

References

[1] Burkhard Rost, *Protein Structure Prediction in 1D, 2D and 3D*

[2] Burkhard Rost, *Rising accuracy of protein secondary structure prediction*

[3] P. Baldi, S. Brunak, P. Frasconi, G. Pollastri, and G. Soda. *Exploiting the past and the future in protein secondary structure prediction.* Bioinformatics, 15:937-946, 1999

[4] G. Pollastri, D. Przbylski, B. Rost, and P. Baldi. *Improving the prediction of protein secondary structure in three and eiqht classes using recurrent neural networks and profiles*

[5] B. Rost and C. Sander. *Improved prediction of protein secondary structure by use of sequence profiles and neural networks.* Proc. Natl. Acad. Sci. USA, Vol.90, pp. 7558-7562, 1993

[6] S. Haykin. *Neural Networks - A Comprehensive Foundation*, Ed.2, AWL, 2001

[7] R. J. Schalkoff. *Artificial Neural Networks.* McGraw-Hill, 1997

[8] D. T. Jones, *Protein secondary structure prediction based on position specific scoring matrices.* J. Mol. Biol.292:195-202, 1999

[9] P. Baldi, S. Brunak, P. Frasconi, G. Pollastri, and G. Soda. *Bidirectional dynamics for protein secondary structure prediction.* Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence(IJCAI99), 1999

[10] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne. *The Protein Data Bank.* Nucleic Acids Research,28, pp. 235-242, 2000 (http://www.rcsb.org/pdb)

[11] U.Hobohm, M.Scharf, R.Schneider, C.Sander. *Selection of a representative set of structures from the Brookhaven Protein Data Bank*, Protein Science 1 , 409-417, 1992

[12] U.Hobohm and C.Sander. *Enlarged representative set of protein structures*, Protein Science 3, 522, 1994

[13] S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman. *Gapped BLAST and PSIBLAST: a new generation of protein database search programs*, Nucleic Acids Res. 25:3389-3402 (http://www.ncbi.nlm.nih.gov/BLAST/)

[14] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. *Basic local alignment search tool*, J.Mol.Biol. 21:403-410, 1990

[15] *Protein Prediction Center, Lawrence Livermore National Laboratory*, California (http://www.predictioncenter.llnl.gov/)

[16] Werbos, P.*Backpropagation Through Time: What it does and How to do it*, Proc.IEEE, Vol.78,No.10,1990

[17] B. Krose, and S. Patrick. *An introduction to Neural Networks.* Ed 8, 1996 (http://www.fwi.uva.nl/research/neuro)

[18] C. Brandsen, and J. Tooze. *Introduction to Protein Structure*, Garland Publishing Inc., 1991