Egyptian Computer Science Journal (ISSN: 1110 – 2586)
Volume 40 – Issue 03, September 2016

The Effect of Using Different Neural Networks Architectures on the Protein Secondary Structure Prediction

Hanan Hendy, Wael Khalifa, Mohamed Roushdy, Abdel Badeeh Salem
Computer Science Department, Faculty of Computer and Information Sciences, Ain-Shams University, Cairo, Egypt
hanan.hendy@cis.asu.edu.eg, wael.khalifa@cis.asu.edu.eg, mroushdy@cis.asu.edu.eg, absalem@cis.asu.edu.eg

Abstract

Knowing that predicting the secondary structure of protein can help deeply in the protein functionality understanding and helps in multiple diseases diagnoses. Moreover, it can help in giving an accurate prediction to the tertiary structures. This forces any researcher to try multiple experiments to find ways to enhance the prediction accuracy. Through the past decade, many machine learning techniques have been used to predict the secondary structures. The main used technique was neural networks. This paper presents three different experiments that make use of artificial neural networks. The first uses a single neural network with different number of hidden layers and nodes. The second combines the output of two neural networks to enhance the accuracy. Last, the third compared to the previous two predicts not only the broad classes of secondary structure (namely; alpha, beta and coil) but predicts all the classes, then combines the result. All the experiments are based on a data set pulled out from the Rcsb protein data bank. The results of all experiments show that the highest accuracy is reached when encoding the primary sequence using binary format and use Feed-Forward network. The accuracy reached around 86% when predicting Beta strands or sheets only. Combining the results of two neural networks showed accuracy of about 83%. Moreover, merging the results of alpha and beta predictors didn't show high significance.

Keywords: *Artificial Neural Networks, Bioinformatics, Machine Learning, Protein Secondary Structure Prediction.*

1. Introduction

Protein secondary structure prediction process involves predicting the main classes of secondary structure which are alpha helix, beta sheets and coil. The prediction process is based on the knowledge of the primary structure a.k.a. amino acid sequence. On the other hand, the tertiary structure prediction [1] is based on predicting the tertiary structure from the primary and secondary structures. Predicting both the primary and the tertiary structures prediction is very important. This importance is because secondary structure prediction helps in diagnosing protein disorder and figuring out diseases. Moreover, accurate prediction of secondary structure helps in high accuracy of the tertiary structure which is important and

known to be hard as well. One of the things that make the tertiary structure prediction difficult is the number of previously known tertiary sequences.

In mid-2011, there were only 70,000 tertiary structures in the PDB [2] -Protein Data Bank - compared to 12.5 million protein sequences in the RefSeq database [3]. So it can be simply seen that the pace by which the primary sequences are added is far large compared to the secondary and huge compared to the tertiary.

Protein forms its secondary structure when the primary amino acids bond together forming one of three shapes. The primary structure as discussed is formed from amino acids. There are twenty main amino acids shown in Table 1 [4]. Table 2 [5] shows the secondary structures named: alpha helix, beta sheets/strands or loops. Their difference depends on the way amino acids bond together.

Table 1 Amino acids primary structures

Polar amino acids			**Polar** amino acids		
Alanine	Ala	A	Aspartic acid	Asp	D
Glycine	Gly	G	Glutamic acid	Glu	E
Valine	Val	V	Arginine	Arg	R
Leucine	Leu	L	Lysine	Lys	K
Isoleucine	Ile	I	Histidine	His	H
Proline	Pro	P	Asparagine	Asn	N
Phenylalanine	Phe	F	Glutamine	Gln	Q
Methionine	Met	M	Serine	Ser	S
Tryptophan	Trp	W	Threonine	Thr	T
Cysteine	Cys	C	Tyrosine	Tyr	Y

Table 2 Protein secondary structure

Amino Acid	**Letter Code**
G, H, I	Alpha helix
T, E, B, S	β-bridge , β-Sheet
B	Residue in isolated
C	Coil

Since proteins are responsible for all the biological tasks in the human body such as generating antibodies that fight infection, hemoglobin that carries oxygen, hormones, enzymes and many more [6], The goal of the secondary structure prediction is to know the bonds that the amino acids form, which will help scientist take the correct actions in case proteins don't fold correctly i.e. don't form their secondary structure the correct way that will result in disorders and diseases.

The predication process involved multiple techniques over the years, Paper [7] contains a detailed comparison of various secondary structure prediction methodologies. It can be seen that long ago statistical and probabilistic techniques were used. The methods used were based on analyzing the residues of the primary sequence getting relative frequencies. The set of probabilities generated are then used in the prediction process. The difference among different techniques of the statistical decade is the use of additional information of primary sequence. For example, some used local interactions others used the conditional probability, etc. Later

machine learning techniques arose that helped in enhancing the prediction accuracy.

Neural networks [8] have been used as the machine learning technique for the secondary structure prediction process. The first use was by Qian and Terrence [9]. They worked on a network with 17 input groups having 21 units per group, 40 hidden units and three output units. They used a dataset of about 106 proteins and reached accuracy of about 64.3%. Later, Chandonia et al [10] used some additional info in the prediction process that helped them reach an accuracy of 73.9% for class prediction using 69 globular proteins (chains). Then a more advanced neural networks were then proposed by Pollastri et al [11]. They used bidirectional recurrent neural network and their accuracy reached 78%. Recently a combined method was introduced by Y. Wei et al [12] in which the prediction is accomplished using the value from multiple predictors, these values are then combined to find out the likelihood of the amino acid sequence. They used a data set of 3000 proteins and reached accuracy reached 83.04%.

In this paper, section 1 presents an introduction about the protein secondary structure prediction, the techniques used and the usage of neural networks specifically in the protein structures prediction. Section 2 shows the data pre-processing in order to be ready for the learning process. Later, section 3 shows the implementation of the neural network and the results obtained comparing the accuracy with the variations of computation parameter. Finally, the conclusion is presented showing what future enhancements can be done and which experiments to be conducted to increase the accuracy.

2. Datasets and Data Pre-processing

One of the main challenges when dealing with protein structures, is the variations of structures and the uncertainty of the sequences. This section presents which data set is used and how the raw files are processed to reach the needed dataset to start the learning process. All the experiments are based on a data set extracted from the PDB [2]. Specifically, the two data sets used are the CB513 [13] and the 396.concise [14] each has separate files each representing a single protein sequence of variable lengths. Both datasets contain variable length protein structures ranging from 20 amino acids to 754. The CB513 is used for both training and testing while the 396.concise is used for testing only.

Each file contains data about a single protein sequence. The aim of preprocessing is to convert these separate files to matrix-like form ready for machine learning process. To start any machine learning algorithm, you need to have a specific input and output. Preprocessing steps are shown briefly in fig. 1 then will be explained in detail. Later a simple example will be explained.

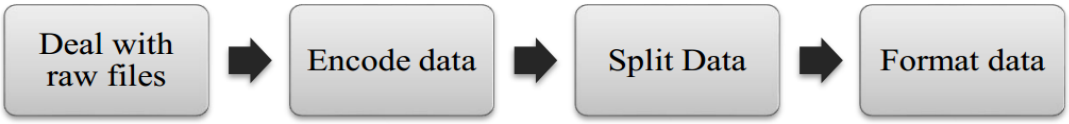

Figure 1 Preprocessing steps of protein data

2.1. Dealing with raw files

The raw file extracted from the data set is represented in FASTA format [15] which is a text-based format for representing either nucleotide sequences or peptide sequences, in which nucleotides or amino acids are represented using single-letter codes. Each file contains multiple data rows that represent different structures and formats for the given sequence. In the conducted experiment only two rows are needed. RES and DSSP. RES represents the protein residue (primary structure) while DSSP represents the secondary structure from the DSSP database.

2.2. Primary and secondary structures separation:

The first step done on these files is collecting all the primary structures and the secondary structures together in two separate files. Then all the commas that separates the characters of the sequence are removed and replace the in the secondary structures with a coil (C).

2.3. Encoding primary and secondary structures:

The next step after having the primary and secondary structures separated is to encode them. The encoding step includes assigning a numeric value for each letter of the input. First the primary structure is represented in one of 20 main amino acids. The ambiguous amino acids like Asx (B), Glx (Z) [16] are dealt with in two different manners, experiments are conducted first when all are encoded as a single value and then the same experiments are conducted with each having a unique value. This is done to check their effect on the accuracy and how significant they are. The encoding of primary structure and secondary structures are shown in table 3 and 4 respectively. After finishing this step, two files are there one having all encoded primary structures and the other having all encoded secondary structures as explained above. In the first two experiments, the secondary structures are encoded to values 01 till 03 while in the third, they are split to their base classes and encoding values ranged 01 till 08

Table 3 - Encoding of protein primary structure

Amino Acid	Letter Code	Encoding Value	Amino Acid	Letter Code	Encoding Value	
Alanine	A	01	Proline	P	13	
Cysteine	C	02	Glutamine	Q	14	
Aspartic acid	D	03	Arginine	R	15	
Glutamic acid	E	04	Serine	S	16	
Phenylalanine	F	05	Threonine	T	17	
Glycine	G	06	Valine	V	18	
Histidine	H	07	Tryptophan	W	19	
Isoleucine	I	08	Tyrosine	Y	20	
Lysine	K	09	Asparagine	B	21	21
Leucine	L	10	Glutamine	Z		22
Methionine	M	11	Leucine	J		23
Asparagine	N	12	Unspecified	X		24

Table 4 - Encoding of protein secondary structure

Secondary Structure	Letter Code	Encoding Value	
Alpha Helix (3-turn helix)	G	01	01
Alpha Helix (4-turn helix)	H		02
Alpha Helix (5-turn helix)	I		03
Beta Sheet	E	02	04
Others (bridge)	B	03	05
Others (bend)	S		06
Others (hydrogen bond)	T		07
Others (coil)	C		08

2.4. Prepare input and output for machine learning phase

Having encoded values for input and output which is the primary and secondary structures respectively is not enough to start the machine learning process. The problem needs to be more specific and aligned. Since the sequences are not of equal lengths and the output structure sequence may have the three secondary structures (alpha helix, beta sheet and coils), a generalization is needed in which the problem is seen as inputting a sequence and deciding whether it's alpha helix, beta sheet or coil. How will this be done? By choosing a number which will specify the number of amino acids from the primary sequence that will be considered a single input. This number is a constant through the learning and testing phases. The corresponding output will be a number representing alpha helix, beta sheet and coil and it will be the one at the mid index of the chosen constant. For example, choose the number 7 which means that the first 7 amino acids will be the input and the output will be the secondary structure mapped to index 4 (as if the amino acid number 4 is being checked when it's in a sequence of 7 amino acids). To be more specific, the goal is to learn the way the N amino acid –here 7– will interact and bond producing the secondary structure at N/2 position.

This step will be clearer later when explained with an example. In this stage, all primary sequences are combined together with "00" separator also the same is done for the secondary sequences. So all the sequences are seen as one sequence. Then this primary sequence is divided with the value chosen before (i.e. 7). So the first seven are taken then the first amino acids are dropped to take the next and so on as shown below. When a sub sequence is found to have "00" in the mid place (i.e. 4th place), the sequence is discarded. This “00” place is discarded as it’s the mapping of the concatenation and not a real amino acids bonding as shown in this example:

151703***02***2006121812150803171706011602091701091 30
1517030***20***06121812150803171706011602091701091 30
15170302***06***12181215080317170601160209170109130

Then the output for each is taken by skipping the first 3 numbers as the first to have a secondary output will be at the 4th position.

Figure. 2 represent an example of the steps explained above.

Raw File

P: RES:A,P,A,F,S,V,S,P,A,S,G,A,S,D,G,Q,S,V,S,V
P: RES:T,P,A,F,N,K,P,K,V,E,L,H,V,H
S: DSSP: , ,E,E,E,E,E, , ,S,S, , ,S,S, ,E,E,E,E
S: DSSP: , ,S, , ,S, ,E,E,E,E,E,E,E

Remove commas

P: APAFSVSPASGASDGQSVSV
TPAFNKPKVELHVH
S: CCEEEEECCSSCCSSCEEEE
CCSCCSCEEEEEEE

Encode

P: 011301051618161301160601160306141618161800171301051209130918041007180717130105120913091804100718071
S: 030302020202020303030303030303030302020202030303030303030303020202020202020202

Combine with 00

P: 011301051618161301160601160306141618161800171301051209130918041007180717130105120913091804100718071
S: 030302020202020303030303030303030302020202000303030303030303020202020202020202

Input Matrix

Split by odd number to make the input and output matrices
In this example, 9 will be used. The matrices are column based

01 13 01 05 16 18 16 13 01 16 06 01 16 03 06 14 18 16 18 00 17 13 01 05 12 09
13 01 05 16 18 16 13 01 16 06 01 16 03 06 14 16 16 18 00 17 13 01 05 12 09 13
01 05 16 18 16 13 01 16 06 01 16 03 06 14 16 18 18 00 17 13 01 05 12 09 13 09
05 16 18 16 13 01 16 06 01 16 03 06 14 16 18 16 00 17 13 01 05 12 09 13 09 18
16 18 16 13 01 16 06 01 16 03 06 14 16 18 16 18 17 13 01 05 12 09 13 09 18 04
18 16 13 01 16 06 01 16 03 06 14 16 18 16 18 00 13 01 05 12 09 13 09 18 04 10
16 13 01 16 06 01 16 03 06 14 16 18 16 18 00 17 01 05 12 09 13 09 18 04 10 07
13 01 16 06 01 16 03 06 14 16 18 16 18 00 17 13 05 12 09 13 09 18 04 10 07 18
01 16 06 01 16 03 06 14 16 18 16 18 00 17 13 01 12 09 13 09 18 04 10 07 18 07

Output Matrix

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 1 1 1
0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 0 0 0

Figure. 2 - Real example for data pre processing

P and S refers to primary and secondary structure files respectively.

3. Implementation and Results

Artificial neural network is chosen to be the machine learning tool used to predict the protein secondary structure. *MTLAB R2013a* neural network tool box is used to implement different neural networks and use different architectures. Implementation varies according to specific computing parameters as shown in table 4. This resulted in 1024 different experiment. Later additional experiments were conducted in which the highest accuracy networks were combined to reach higher prediction accuracy. This will be discussed after discussing the main results.

Table 4 - Computation parameters

Computing Parameter	Different Values for the parameter
Neural Network used Architectures: - Feed Forward - Back propagation	
Number of input neurons - The input file that represents the primary structures was split to form the input matrix using different numbers. This decides the number of amino acids that will be given to the neural network at once. o 17,19,25 and 31 Input neurons	
Number of Hidden Layer and Neurons per layer - Single hidden layer with different number of neurons. (10 or 3 neurons) - Multiple hidden layer with different number of neurons o Two hidden layers with 10 neurons and 3 neurons respectively. o Two hidden layers with 3 neurons and 10 neurons respectively.	
Input format - **Integer** input with ignoring ambiguous amino acids: such that amino acids are encoded as numbers from 1 to 21 - **Integer** input without ignoring ambiguous amino acids: such that amino acids are encoded as numbers from 1 to 24 - **Binary** input: such that amino acids are encoded as binary values (0 and 1). Each amino acid is encoded to a string of 0's and 1 at the index of the amino acid. For example: sequence 0307 will be encoded to 00100000000000000000 00000010000000000000	
Prediction Expected Output Classes - Predict alpha and beta together - Predict alpha only - Predict beta only - Predict alpha and beta alone and merge the results	
Percentage of data used for training, testing and validation - **70 : 15 : 15 - 80 : 0 : 20 - 75 : 0 : 25**	

3.1. Experiment One

As mentioned earlier, 1024 experiment are conducted and for each the number of wrongly predicted secondary structures were counted. This resulted in having an accuracy level for each experiment which helped in concluding the findings discussed later in this section.

First, table 5 shows the accuracy resulted when having 70% of the data for training, 15% for validation and 15% for testing. Ambiguous amino acids were encoded as one value (i.e. 21 values to differentiate amino acids). Table 6 conducts the same experiment but using backpropagation neural network. The best accuracy reached in the first experiment was 62% when using NN with two hidden layers the first with 3 neurons and the second with 10 and predicting beta only using binary format for encoding. While the worst was reached when predicting alpha and beta each alone and merging the output. A two layer NN was used with 3 neurons in the first layer and 10 in the second. The accuracy in this case was 44.69%.

The second experiment showed the best accuracy and worst accuracies with the same NN. The one with two hidden layers with 3 and 10 neurons respectively. The best was 82% when predicting beta only and the worst was 44.68% when predicting both and merging the output same as first experiment.

Table 7 shows the result of distributing the data as 70% training, 15% validation and 15% testing using feedforward but this time without ignoring the ambiguous amino acids (i.e. 21 - 24). The backpropagation experiment results are shown in table 8. The best accuracy in these two experiments were when predicting beta only with binary format. The accuracy reached 86.26% in the first and 84.52% in the second. The first used a two layer NN with 10 and 3 neurons respectively while the second used a 10 neuron single layer NN. The worst accuracy in the first case was 46.76% and 46.52 in the second. Both experiments tend to predict alpha and beta each alone then merge the output. The 46.76% was reached when using a single layer NN with 100 neurons and the 46.52 was reached with a two layer NN with 10 and 3 neurons respectively.

Table 5 - Prediction Accuracy (FF Neural Network, No ambiguous amino acids)

Encoding (Numeric, Binary)	Prediction of alpha only, beta only, both, merged	One Hidden Layer with 10 nodes	One Hidden Layer with 3 nodes	Two Hidden Layers with 10 and 3 nodes respectively	Two Hidden Layers with 3 and 10 nodes respectively
Numeric	Both	45.81%	45.49%	45.41%	45.37%
Binary	Both	64.01%	58.68%	63.54%	58.78%
Numeric	Alpha	65.81%	65.76%	65.70%	65.74%
Numeric	Beta	78.75%	78.75%	78.75%	78.76%
Binary	Alpha	75.67%	75.63%	75.63%	75.38%
Binary	Beta	81.99%	81.79%	81.58%	**82.09%**
Numeric	Merged	44.97%	44.86%	44.69%	44.88%
Binary	Merged	61.26%	61.14%	60.31%	61.06%

Table 6 - Prediction Accuracy (BP Neural Network, No ambiguous amino acids)

Encoding (Numeric, Binary)	Prediction of alpha only, beta only, both, merged	One Hidden Layer with 10 nodes	One Hidden Layer with 3 nodes	Two Hidden Layers with 10 and 3 nodes respectively	Two Hidden Layers with 3 and 10 nodes respectively
Numeric	Both	45.72%	45.29%	45.25%	45.54%
Binary	Both	63.70%	63.81%	64.14%	59.11%
Numeric	Alpha	65.69%	65.69%	65.71%	65.74%
Numeric	Beta	78.73%	78.76%	78.75%	78.76%
Binary	Alpha	75.16%	75.37%	75.51%	75.55%
Binary	Beta	81.86%	81.91%	**82.08%**	81.95%
Numeric	Merged	44.77%	44.86%	44.68%	44.85%
Binary	Merged	61.22%	61.07%	61.43%	61.62%

Table 7- Prediction Accuracy (FF Neural Network, unique encoding for ambiguous amino acids)

Encoding (Numeric, Binary)	Prediction of alpha only, beta only, both, merged	One Hidden Layer with 10 nodes	One Hidden Layer with 3 nodes	Two Hidden Layers with 10 and 3 nodes respectively	Two Hidden Layers with 3 and 10 nodes respectively
Numeric	Both	48.46%	47.21%	48.51%	49.16%
Binary	Both	65.30%	64.71%	66.64%	66.98%
Numeric	Alpha	66.40%	66.41%	66.15%	66.15%
Numeric	Beta	79.14%	79.17%	79.30%	79.13%
Binary	Alpha	80.25%	80.61%	81.32%	80.94%
Binary	Beta	83.96%	84.05%	**86.26%**	84.03%
Numeric	Merged	46.76%	46.90%	46.81%	47.00%
Binary	Merged	67.26%	67.53%	69.99%	67.66%

Table 8 - Prediction Accuracy (BP Neural Network, unique encoding for ambiguous amino acids)

Encoding (Numeric, Binary)	Prediction of alpha only, beta only, both, merged	One Hidden Layer with 10 nodes	One Hidden Layer with 3 nodes	Two Hidden Layers with 10 and 3 nodes respectively	Two Hidden Layer with 3 and 10 nodes respectively
Numeric	Both	48.74%	48.55%	47.60%	47.16%
Binary	Both	66.23%	64.47%	65.59%	66.10%
Numeric	Alpha	66.47%	66.57%	66.18%	66.33%
Numeric	Beta	79.29%	79.25%	79.15%	79.29%
Binary	Alpha	80.23%	80.22%	77.86%	78.55%
Binary	Beta	**84.52%**	84.30%	83.86%	82.89%
Numeric	Merged	46.96%	47.37%	46.52%	47.04%
Binary	Merged	67.35%	65.91%	64.49%	64.12%

The following observations where made based on all the conducted experiments:

- Numeric (integer) encoding for input showed less accuracy than binary encoding by about 15 to 20 percent.
- Predicting the alpha or the beta secondary structure is better than predicting both together or predicting each and then combining the results.
- Beta structure prediction through all the experiments shows the highest accuracy with all varying parameters.

As shown above some detailed results were discussed comparing results among different architectures and computing parameters. Tables 9 through 11 show the most and least error rates among all experiments. It is clearly seen that the highest error rate results when predicting both alpha and beta together using input in number format ranging from 50% to 56% with accuracy about 45% - 50%. The least accuracy is shown in table 9 in detail. This again proofs the discussed output above; the numeric formatting should not be used in the prediction process. While the lowest error rate results when predicting Beta structures alone with binary representation for the input. Error rate in this case range from 14% to 18% with accuracy about 85% - 86% as shown in table 10. Table 11 shows the average performance when predicting alpha using binary format. Error rate average is 35% with accuracy 64% - 68%. Above average accuracy is represented in table 12. This experiment is predicting the alpha or beta each alone and as the highest accuracy, binary format was used. The accuracy ranged from 75% to 81% and error rate 25% to 19%.

Table 9 - Accuracy when predicting alpha and beta using number format (Worst accuracy)

	Back propagation network				**Feed forward network**			
	Number of hidden layers							
Number of input nodes	**10**	**3**	**10 3**	**3 10**	**10**	**3**	**10 3**	**3 10**
17	47.85%	45.67%	47.20%	46.66%	48.76%	48.99%	49.00%	49.16%
19	45.81%	46.18%	45.59%	46.31%	46.16%	46.58%	46.30%	46.37%
25	48.74%	48.55%	47.60%	47.16%	48.46%	49.04%	49.06%	47.59%
31	45.40%	45.56%	46.40%	45.31%	48.59%	46.45%	46.57%	46.43%

Table 10 - Accuracy when predicting beta using binary format (Best accuracy)

	Back propagation network				**Feed forward network**			
	Number of hidden layers							
Number of input nodes	**10**	**3**	**10 3**	**3 10**	**10**	**3**	**10 3**	**3 10**
17	82.23%	82.24%	83.13%	82.16%	83.08%	83.35%	83.73%	82.91%
19	82.00%	81.96%	82.08%	82.89%	82.82%	82.04%	82.63%	82.36%
25	84.52%	84.30%	83.86%	82.22%	85.07%	84.05%	**86.26%**	84.03%
31	81.99%	82.04%	82.22%	82.31%	82.09%	82.47%	82.48%	83.03%

Table 11 - Accuracy when predicting alpha using binary format (average accuracy)

	Back propagation network				Feed forward network			
	Number of hidden layers							
Number of input nodes	10	3	10 3	3 10	10	3	10 3	3 10
17	66.00%	63.81%	65.59%	64.88%	65.80%	64.71%	65.75%	65.83%
19	64.19%	63.92%	63.54%	64.95%	64.09%	59.79%	64.40%	59.53%
25	66.23%	64.47%	64.67%	66.10%	65.80%	65.57%	67.84%	66.98%
31	64.01%	63.96%	64.64%	64.14%	64.17%	64.55%	64.66%	64.53%

Table 12 - Accuracy when predicting alpha using binary format (above average accuracy)

	Back propagation network				Feed forward network			
	Number of hidden layers							
Number of input nodes	10	3	10 3	3 10	10	3	10 3	3 10
17	78.02%	80.22%	77.86%	76.73%	79.65%	80.03%	78.39%	78.43%
19	75.45%	75.70%	75.51%	76.06%	75.61%	75.74%	75.94%	77.53%
25	80.23%	78.50%	75.84%	78.55%	81.02%	80.61%	81.32%	80.94%
31	75.77%	77.36%	75.94%	76.09%	76.94%	75.95%	76.84%	76.18%

3.2. Experiment Two

After analyzing all the above results, two networks were combines to reach a higher accuracy for alpha prediction and beta prediction separately. The idea is based on the likelihood from both networks. The input sequence is predicted by both networks and the output is combined. If any of the two networks predicts the sequence as alpha then it is alpha and if beta then it is beta. Figure 3 shows the chosen networks for merged prediction. Then table 13 shows the result difference with and without combining the network. Last the experiment is tested using 396.concise and the results are seen in table 13 as well.

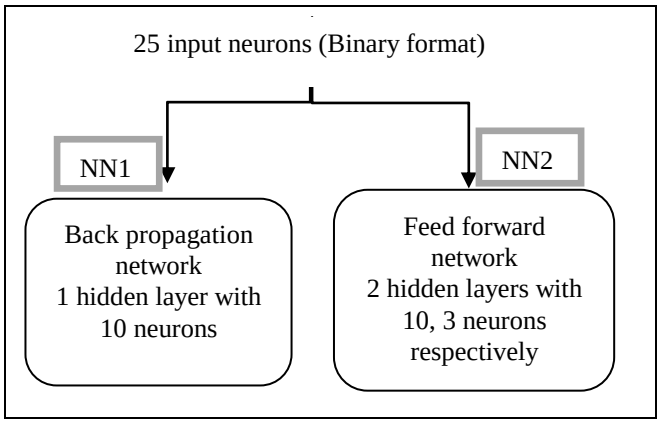

Figure 3 – Merged NN used in prediction process

Table 13 - Accuracy of combined NNs

Dataset		CB513	396.concise
Alpha	NN1	75.23%	70.28%
	NN2	74.89%	70.13%
	Combined	**77.70%**	**72.91%**
Beta	NN1	81.78%	76.22%
	NN2	81.45%	76.05%
	Combined	**83.36%**	**77.85%**

It can be concluded from table 13 that combining NN increases the accuracy by about 2%. Comparing this experiment result with other results that used the same machine learning technique, the following is concluded. The results exceed the accuracy of John et al [17] whose results had an accuracy ranging from 62.3% to 73.9%. Also exceeds those of Qian and Terrence [9] who reached 64.3%, Chandonia et al [10] who reached almost 73.9 and Pollastri et al [11] who reached around 78%. Moreover, the most recent method presented by Y. Wei et al [12] showed the best accuracy when using Neural Networks which approached 83.04%.

3.3. Experiment Three

The third experiment is conducted using the secondary structure encoded in their base classes without combining any. Thus the difference between the first and the third experiment is with the prediction expected output. Two sub experiments were conducted. The first compares the output of prediction without post processing. The second, post process the output to combine them back to the three classes only. Table 14 shows the results of both experiments.

Output post-processing	No of input nodes	Input Encoding	One Hidden Layer with 10 nodes	One Hidden Layer with 3 nodes	Two Hidden Layers with 10, 3 nodes	Two Hidden Layers with 3, 10 nodes
No post processing	17	Numeric	53.62%	53.70%	53.73%	53.56%
	17	Binary	52.01%	**63.06%**	**60.41%**	56.87%
	19	Numeric	53.70%	53.61%	52.02%	53.51%
	19	Binary	**63.42%**	58.40%	52.01%	52.01%
	25	Numeric	53.79%	53.64%	53.68%	52.84%
	25	Binary	**60.34%**	52.01%	59.49%	52.01%
	31	Numeric	53.77%	53.87%	52.01%	53.55%
	31	Binary	59.83%	52.01%	59.07%	52.32%
Merge back all output to 3 classes (Alpha, Beta and Coil)	17	Numeric	57.01%	56.93%	56.98%	52.01%
	17	Binary	55.73%	**70.89%**	64.56%	55.75%
	19	Numeric	57.24%	56.88%	55.86%	56.80%
	19	Binary	**70.75%**	63.39%	55.72%	55.72%
	25	Numeric	57.11%	56.91%	56.83%	56.36%
	25	Binary	66.80%	55.72%	61.40%	55.72%
	31	Numeric	57.20%	57.03%	55.72%	56.61%
	31	Binary	62.52%	55.72%	62.48%	55.84%

Compared with tables 5, 6 and 7, it can be seen that predicting the three classes didn't exceed 66%. While in this experiment it reached almost 70% with an increase of about 4% - 6%.

4. Conclusion and future work

As discussed all through the paper, artificial neural networks were used to predict the protein secondary structure. The method showed varying accuracy. It is clearly seen that the methods reached an accuracy of 81% for alpha 86% for beta and around 70% for predicting both. This is considered significant with the use of Neural Network. This increase in the accuracy is due to the following:

- Unifying the length of the input while having different protein primary structure lengths.
- Encoding the input and the output in binary format. This made the network either take the whole component or discard it which is different that the case of numeric encoding.

Having reached 86% accuracy is not enough. Further enhancements can be done to increase the accuracy. Some of these enhancements are:

- Combining the results from more than one network and take the highest weight. This way more than one network is taken into consideration.
- Combining results from more than one network but with weighted approach. The network that has as standalone higher accuracy will take a higher weight than the one with less accuracy.
- Using other encoding techniques and different ways to format the input matrix.
- Using more than one machine learning technique and combine the results. By doing this, a better accuracy can be reached that will help later on in the tertiary structure prediction. This is considered the trend in the prediction process these days.
- Using the full encoding of secondary structure in one of the combined networks.

References

[1]. Yosr Eman, Amr Badr and Ibrahim Farag. “Cellular evolutionary algorithms for solving protein folding problem”. Egypt Computer Science Journal, Volume 34, Number 2, 2010.

[2]. Rcsb protein data bank." http://www.rcsb.org/pdb/home/home.do. [Accessed: July - 2015].

[3]. A. E. Kister, Protein Supersecondary Structures, Humana Press, 2013.

[4]. Rodolfo O. Esquivel, Moyocoyani Molina-Espíritu, Frank Salas, Catalina Soriano, Carolina Barrientos, Jesús S. Dehesa and José A. Dobado, "Decoding the Building Blocks of Life from the Perspective of Quantum Information," Advances in Quantum Mechanics, pp. 641-669, 2013.

[5]. Ron Milo and Rob Phillips “Cell Biology By The Numbers” Ch. 3 July 2015.

[6]. The Art of protein structure prediction “http://www.eurekalert.org/features/doe/2004-12/ddoe-tao122204.php” [Accessed : January 2015].

[7]. Hanan Hendy, Wael Khalifa, Mohamed Roushdy and Abdel Badeeh Salem “A Study of Intelligent Techniques for Protein Secondary Structure Prediction” International Journal "Information Models and Analyses" Volume 4, Number 1, p.p. 3-12 2015.

[8]. S. Haykin, Neural Networks and Learning Machines, Ch. 1. Pearson International Edition, 2009.

[9]. Ning Qian and T. J. Sejnowski, "Predicting the secondary structure of globular proteins using neural network models," Journal of Molecular Biology, vol. 202, no. 4, p. 865–884, August 1988.

[10]. John-Marc Chandonia and Martin Karplus, "Neural networks for secondary structure and structural class predictions," Protein Science, vol. 4, no. 2, p. 275–285, February 1995.

[11]. Gianluca Pollastri, Darisz Przybylski, Burkhard Rost and Pierre Baldi, "Improving the Prediction of Protein Secondary Structure in Three and Eight Classes Using Recurrent Neural Networks and Profiles," PubMed, pp. 228-235, May 2002.

[12]. Y. Wei, J. Thompson and C. A. Floudas, "CONCORD: a consensus method for protein secondary structure prediction via mixed integer linear optimization," Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science, pp. 831-850, November 2011.

[13]. "Cuff and barton data set." http://comp.chem.nottingham.ac.uk/disspred/datasets/CB513[Accessed: July - 2015].

[14]. Concise dataset "http://www.compbio.dundee.ac.uk/jpred/downloads/396.concise.tar.gz" [Accessed: July – 2015]

[15]. format." https://en.wikipedia.org/wiki/FASTA_format. [Accessed: June - 2015].

[16]. Amino acids." https://en.wikipedia.org/wiki/Amino_acid. [Accessed: May - 2015].

[17]. J.-M. Chandonia and M. Karplus, "Neural networks for secondary structure and structural class prediction," Cambridge university press, 1995.