Protein Structure Prediction Using Neural Networks

A Literature Review

Martha Mercaldi
Kasia Wilamowska
CSE 527
December 16, 2003

1 Introduction

1.1 History of Protein Folding

Understanding how a strand of amino acids folds to form a three dimensional structure is a big problem whose solution offers similarly big rewards. A breakthrough in this area has the potential to revolutionize medical research, molecular biology and related fields. For 40 years scientists have worked to discover how a protein's amino acid sequence determines its function, and while great strides have been made, the ultimate goal has not yet been achieved.

Proteins are the machinery via which cells perform nearly all of their functions. Because protein interacts physically according to the "key hole" principle, its three dimensional shape determines its behavior and interactions with cellular structures. All information about a protein's 3D shape is encoded in a 1D strand of amino acids which, in solution, forms various secondary structures such as helices, ribbons and loops which further fold up on themselves to take on a distinct 3D form.

Researchers have applied every imaginable pattern discovery, recognition, matching technique to this problem, with several yielding encouraging results. One such approach called comparative modeling has offered fairly accurate predictions of 3D structure by identifying proteins with similar sequences and known structure. Other approaches have tried to break the complex problem down into simpler ones addressing factors such as secondary structure, solvent accessibility and inter residue distances.

Research has shown that similarly structured proteins perform similar functions, so while it is desirable to define functions as precisely as possible, classifying proteins into functional families another intermediate yet informative problem. This classification problem can be approached from many angles, one of which, neural networks, we examine here.

1.2 Evolution of Neural Networks

Neural networks are a machine learning technique originally designed to emulate the behavior of neurons in the brain. Applications of neural networks range from signal processing to stock market predictions, and of course classification as described here.

The most basic neural network is a single layer "feed forward" network. Each input node takes a binary value and the network calculates the output value (between 0 and 1) as a function of the inputs and the parameters on each edge. A simple example is shown in the diagram below.

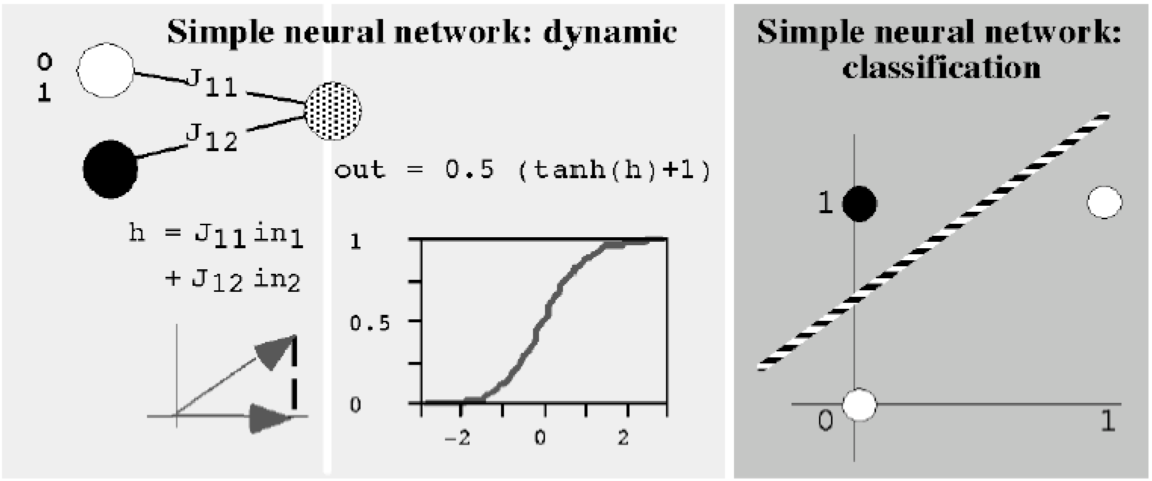


Figure 1.1. Example single layer neural network

Neural networks can be tailored to more complex classification problems by adding additional layers of nodes. In the example below the extra layer allows the classifier to distinguish a more complex pattern of nodes.

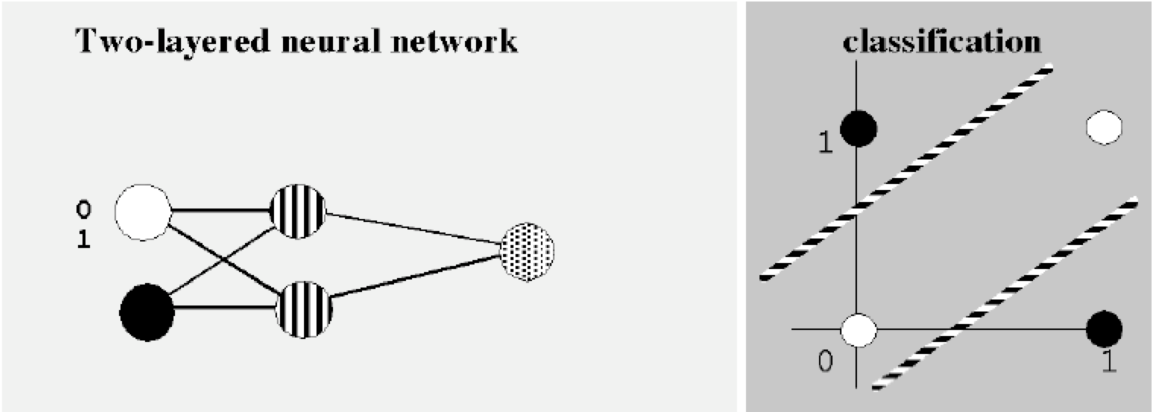


Figure 1.2. Simple multilayer neural network

Naturally the value of the parameter along each edge is critical to the network's function, and these parameters must be learned via training data. Any algorithm that minimizes the network error $(error = (output - desired)^2)$ can be used to select parameters. If any a priori knowledge is available it should be used to make an initial parameter estimate as training can otherwise be time consuming.

The key property of neural networks is that they can learn to recognize patterns, but at the same time generalize and recognize similar patterns that may not have been included in the training set.

2 Why Apply Neural Networks to Protein Folding?

Protein functional families have been shown to have strong amino acid sequence conservation and therefore pose a natural classification problem. In fact, the genetic code by which nucleotide triplets code for amino acids was discovered using a neural network. In the past, when neural networks have been applied to related problems, the results have been promising. Burkhard Rost cites several such efforts in his article "Neural Networks predict protein structure: hype or hit?" published in 2003 in "Artificial Intelligence and Heuristic Methods in Bioinformatics."

Neural networks have also been used to try to deduce the tertiary structure of some proteins by classifying amino acid residues as either "exposed" or "buried" with respect to the molecule's hydrophobia/hydrophilia in solution. Neural networks have even helped detect errors in protein databases. Researchers were able to do this by tracking which test cases always remained unclassifiable, even when the networks were overfitted to the data.

Even with these advances, neural networks have shown signs of even more effective application. As a technique it is able to incorporate evolutionary information in the form of a multiple sequence alignment. This insight pushed secondary structure prediction (classifying secondary state as either a helix, strand or loop) across the apparent 70% accuracy limit to where it stands today at 76%. Multiple sequence alignments were able to improve the results of other experiments as well such as solvent access classification.

Other efforts to inject biological knowledge into the classification process with neural networks have been successful. For example, researchers noticed that in the secondary structure classification experiments, the strands were never particularly well predicted compared to the other two structures. Initially this was attributed to features that were outside of the window frame over which each prediction was made, however it was later noticed that it was the network took about one tenth the time to learn to recognize helices and loops than it took to learn to identify a strand. By adjusting the training data to increase the frequency of strands, researchers were able to train the network to detect strands as effectively as it had been predicting the other two. What is encouraging about this is that the problem was not due to an inherent shortcoming of the model or to long range interactions between acid residues, but rather a technical problem in the way it was used.

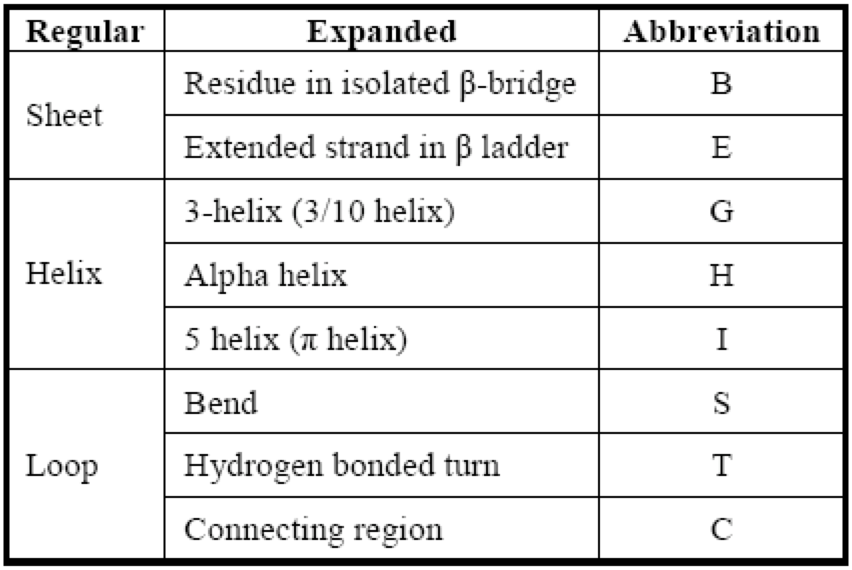
Researchers hope to build on these experiences and push neural networks farther towards exposing comprehensive information about an unknown protein's function. Two such efforts are detailed in the next section.

3 Two examples

3.1 *Protein Secondary Structure Prediction Based on Denoeux Belief Neural Network*

3.1.1 Purpose

The goal of this work was to use neural nets to effectively predict the secondary structure of proteins. Currently the best secondary structure prediction method is SSpro8 with accuracy in the range of 62-63%. Inaccuracies are due to complexity factors used to determine structure conformation. As input to the system, one can choose to use DNA or amino acid sequences. SSpro8 uses amino acid sequences, whereas the authors' system, UTMPred, uses DNA. As output classes for both of the above systems, the forms consisting of alpha helices, beta sheets and loops are expanded to eight structure forms (Table 3.1).



Regular	Expanded	Abbreviation
Sheet	Residue in isolated β-bridge	B
	Extended strand in β ladder	E
Helix	3-helix (3/10 helix)	G
	Alpha helix	H
	5 helix (π helix)	I
Loop	Bend	S
	Hydrogen bonded turn	T
	Connecting region	C

Table 3.1. Protein Secondary Structure Forms

3.1.2 Methodology

At its core the problem of predicting secondary structures in protein is a classification problem. The Denoeux belief neural network (DBNN) model (Figure 3.1) is incorporated within the UTMPred as the classification engine.

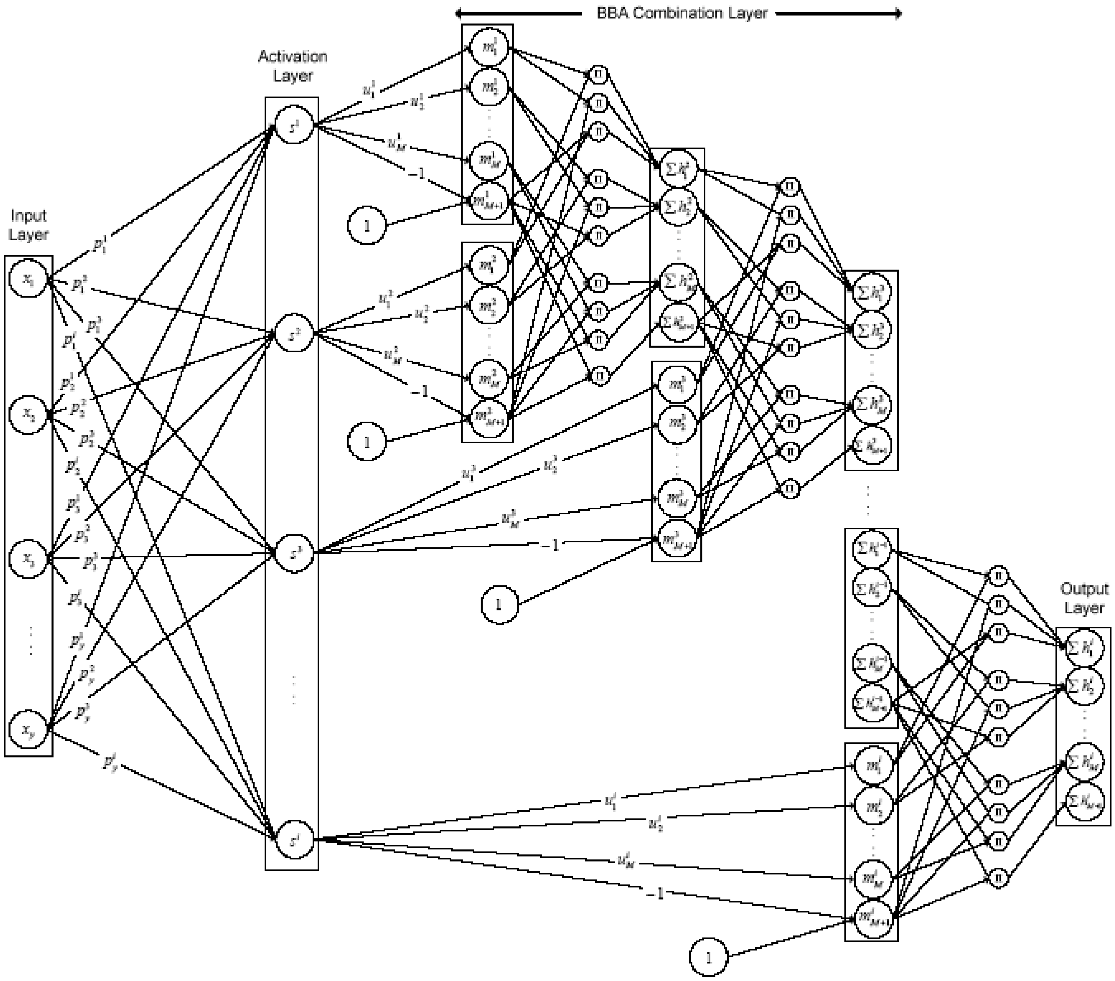


Figure 3.1. Denoeux Belief Neural Network (DBNN) Architecture

The DBNN input are DNA sequences converted to binary format prior to use. A is represented as 1000, C as 0100, G as 0010, and T as 0001.

88 *Escheichia coli* proteins, 25 yeast *Saccharomyces cerevisiae* proteins and 166 mammalian proteins (80 of which are human), are used in the experiment. All of the structures listed above have a resolution greater than or equal to 2.5 _. The input window size for UTMPred is set to 7 codons (Figure 3.2), which results in 84 input nodes and 8 output nodes which represent the structure forms.

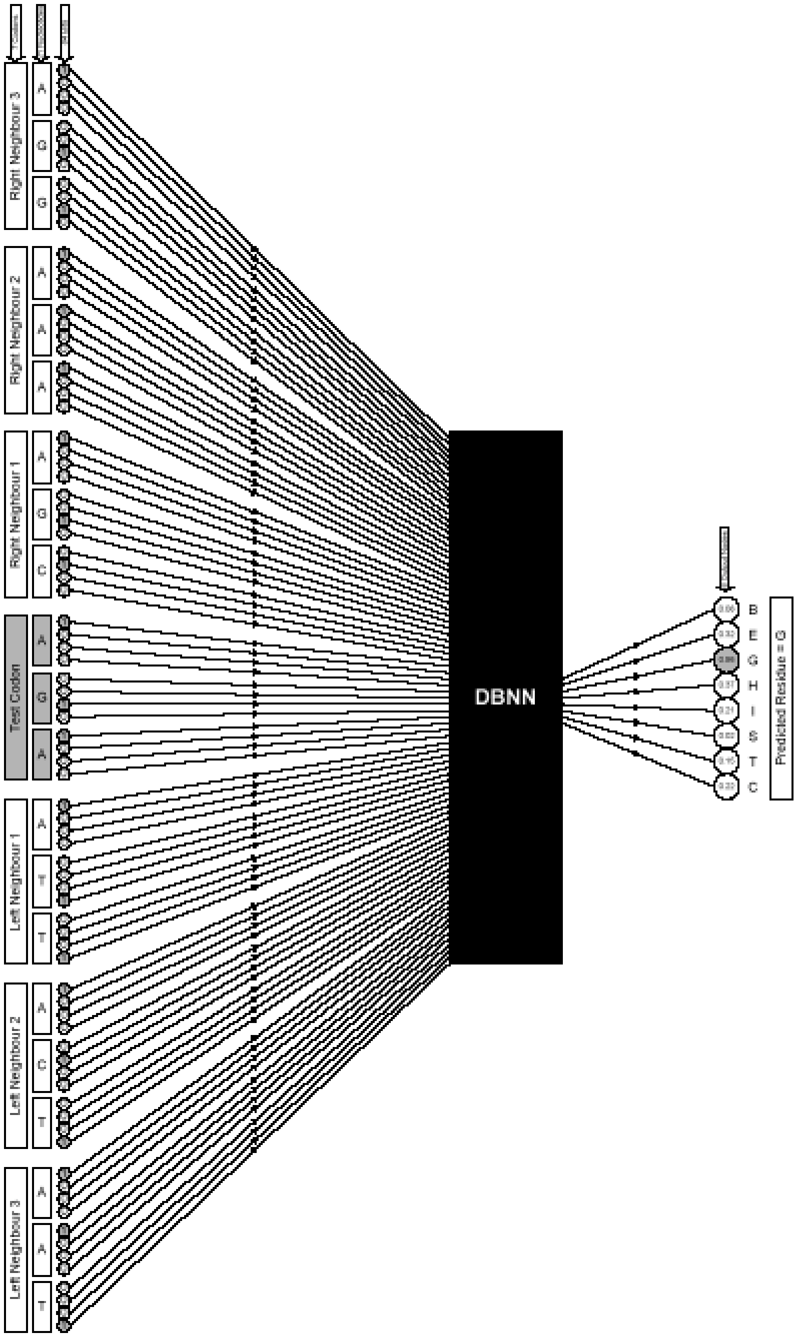
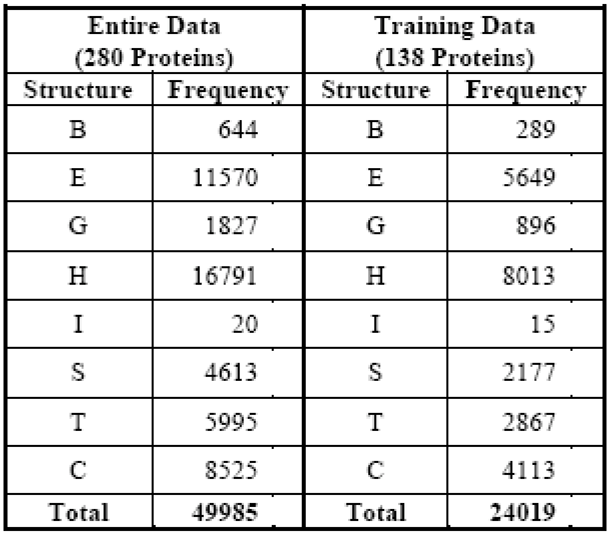


Figure 3.2. Detailed Architecture of UTMPred

3.1.3 Results

UTMPred used 200 prototypes as training data and once the training was complete, the system was able to predict H and E forms with greater than 75% accuracy. At the same time, the system had difficulty predicting form I (5 helix or π helix), due to a small amount of data in the training samples (Figure 3.3a, 3.3b).



Entire Data (280 Proteins)		Training Data (138 Proteins)	
Structure	Frequency	Structure	Frequency
B	644	B	289
E	11570	E	5649
G	1827	G	896
H	16791	H	8013
I	20	I	15
S	4613	S	2177
T	5995	T	2867
C	8525	C	4113
Total	49985	Total	24019

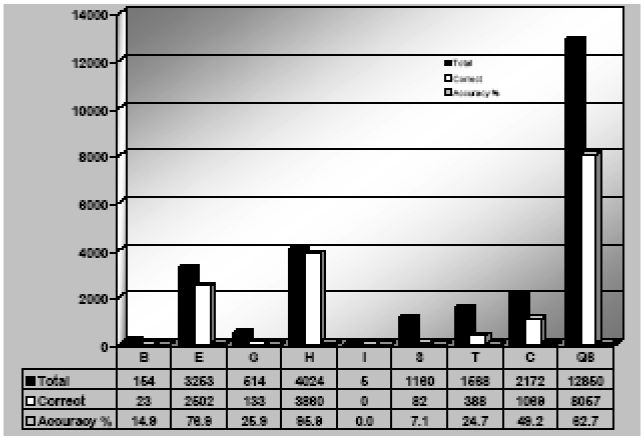


Table 3.3.a. UTMPred Prediction Results

Figure 3.3.b. Statistics with Input Windows Size 7

3.2 Assignment of Protein Sequence to Functional Family Using Neural Network and Dempster-Shafer Theory

3.2.1 Purpose

In this project the previous work was expanded to try to efficiently predict protein function. This can be done either by querying databases such as Prosite, Pfam, and Prints, for motifs within a single protein or to query for an absence or presence of arbitrary combinations of motifs.

3.2.2 Methodology

Given a training set, the task is to induce a classifier that is able to assign novel protein sequences to one of the protein families represented in the training set. Protein sequences with known function are divided into training and testing datasets and a DBNN is used to build a classifier on the training set. Once trained, the classifier will be able to predict novel proteins into specific functional families based on what it learned from the training set (Figure 3.4).

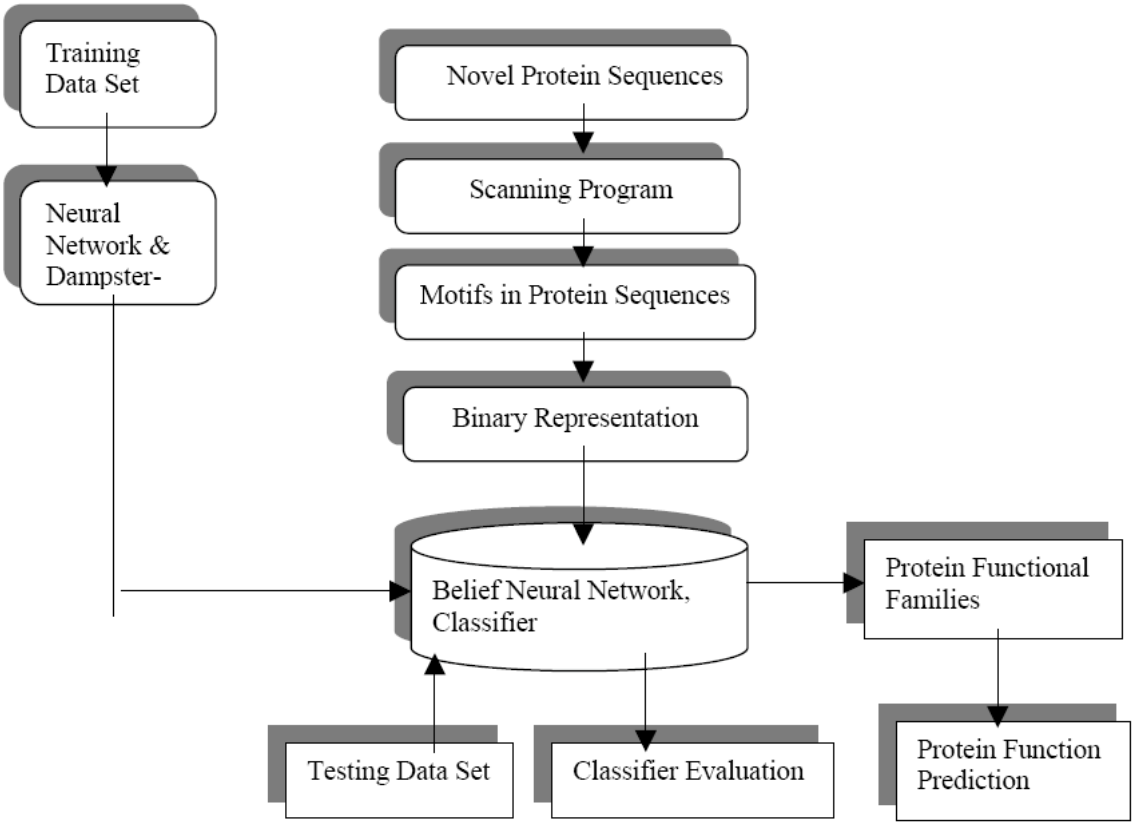


Figure 3.4. Basic approach of protein sequence assignment to functional families

Training and test data consisted of over 1100 entries from the Prosite database. Each entry describes a function shared by some proteins. In the experiment, one Prosite documentation entry corresponded to a protein class, and each protein class could , in turn, be characterized by one or more motif patterns/profiles. Only motifs considered significant matches by profileScan were chosen, and a DBNN was used as the classifier.

3.2.3 Results

The performance of the system was compared to existing statistical and neural network techniques and proved robust to strong changes in the distribution of the input data. Since protein functional predictions is difficult due to the volatility of input data, this robustness is considered extremely useful.

585 proteins belonging to one of ten classes were used, from which subsets of varying size were picked randomly to become the training set. For each set (of size 11, 20, 29, 58, 117, 175, 234, 294, 351, and 585) the experiment was run three times, each time using a randomly sampled training set of the given size. Once the DBNN was trained, all 585 proteins were used as the test set to determine accuracy (Figure 3.5).

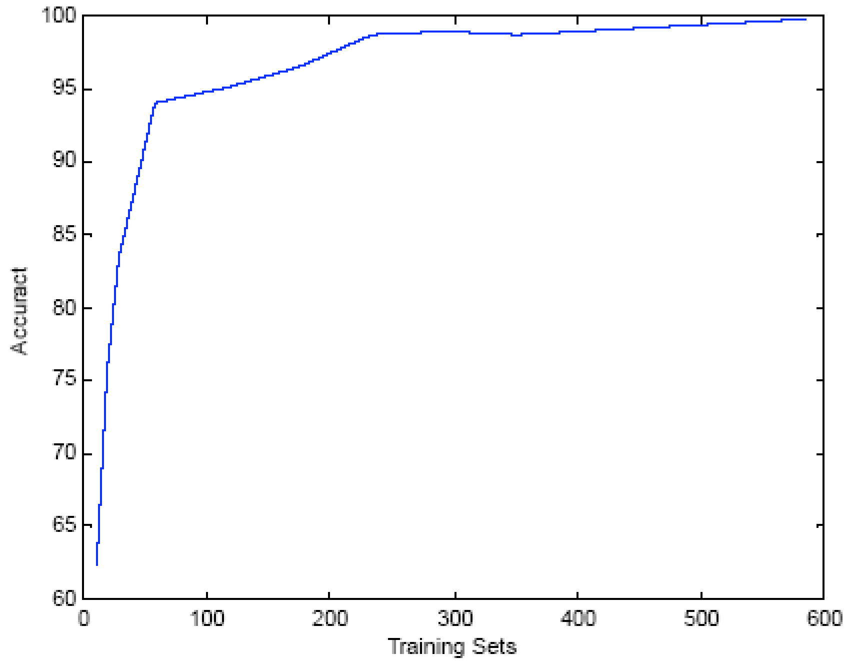


Figure 3.5. With only 10% of the total training samples, DBNN could be constructed to classify proteins with a 95% accuracy.

An additional benefit of the DBNN was that the number of false positives generated were significantly lower than those resulting from a Prosite search. (The original paper [Zaki2003] offers no more information about the search method that was used.) As seen in Figure 3.6, as the size of the data set approaches 100% of the 585 total proteins in the experiment, the false positives discovered by DBNN approaches zero.

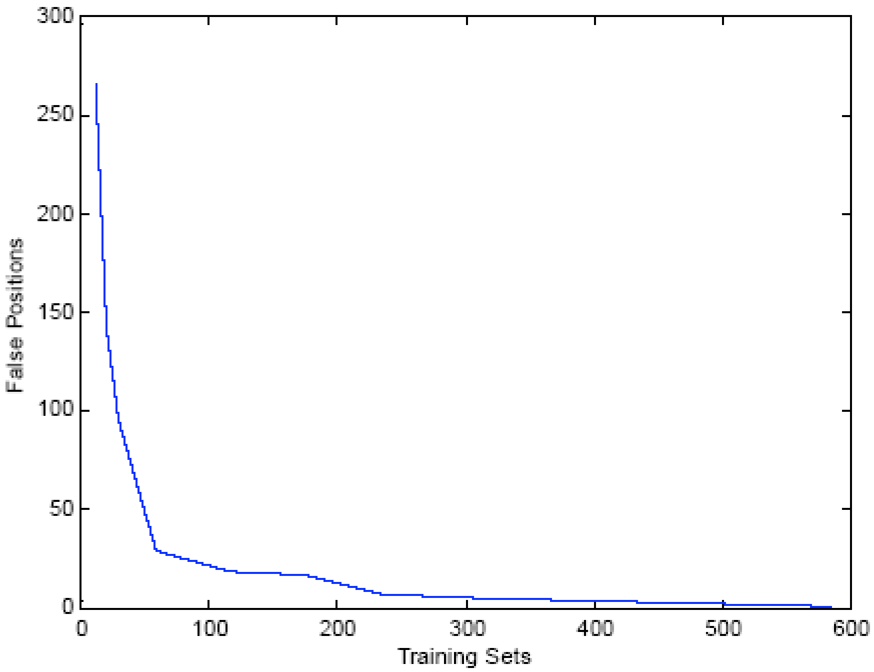


Figure 3.6. The number of false positives resulting from the use of the DBNN trained using training sets of different sizes.

As an additional exploration, a second data set of 73 protein sequences drawn from five classes were used to build a DBNN classifier. These proteins were chosen so that there was significant overlap in motif composition among the families. If querying were to be done in Prosite, there would be a high rate of false positives. However, using the DBNN classifier built by random sized datasets, the output exceeded 96% accuracy when trained on more than 22 data samples. Once the input contained more than 80% (58 or more sequences) of the dataset, all sequences were correctly predicted (Figure 3.7).

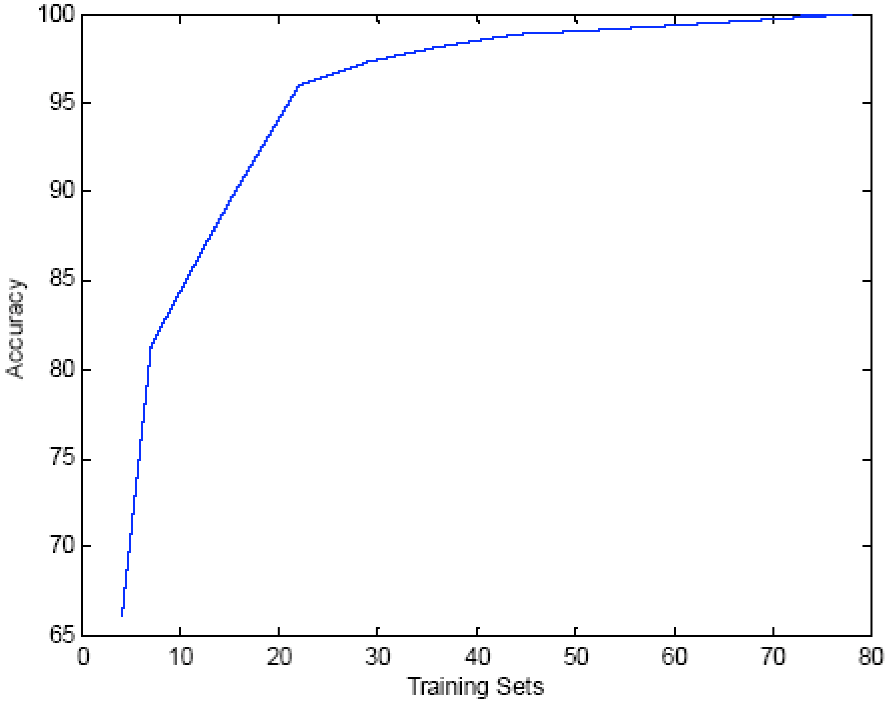


Figure 3.7. Result of classifying proteins containing common motifs.

4 Future Work

It is unlikely that a single discovery will provide the breakthrough to full protein structure prediction. What is more plausible is that the successful approach will incorporate several methods in concert. Neural networks are likely to be part of this solution as they have been shown to effectively analyze nearly any feature relevant to protein function from secondary structures to solvent access to the distances between residues in the final structure. Furthermore, neural networks can combine knowledge from multiple sources, which naturally would be a critical component in this expected hybrid solution.

5 Annotated Bibliography

B. Rost. “Neural networks for protein structure prediction: hype or hit?” Artificial intelligence and heuristic methods for bioinformatics (2003): 34-50.

This paper was an interesting and readable review of past and recent efforts to apply neural networks to the protein folding problem.

S.N.V. Arjunan, S. Deris, R.M. Illias. “Protein Secondary Structure Prediction Based on Denoeux Belief Neural Network.” ICAIET Proceedings (2002): 554-560.

This paper detailed an effort to predict protein secondary structures directly from the DNA sequence. While the results were impressive the paper was not particularly clear and frequently left us with questions.

N.M.Zaki, S. Deris, S.N.V. Arjunan. “Assignment of Protein Sequence to Functional Family Using Neural Network and Dempster-Shafer Theory” Journal of Theoretics 5-1 (2003).

Kasia, I didn’t read this one so well. Will you sum it up?

5.1 Background Information

S.N.V Arkimam, S. Deris, R.M.Illias. “Prediction of Protein Secondary Structure” Jurnal Teknologi 35(C) (2001): 81-90.

T.Wessels, C.W. Omlin.”Refining Hidden Markov Models with Recurrent Neural Networks”.