Extending the Decision Accuracy of a Bioinformatics System

A. Chong, T.D. Gedeon and K.W. Wong

School of Information Technology
Murdoch University
South St.
Murdoch, Western Australia 6150
Australia

Summary: We introduce a simple fuzzy technique to improve the prediction decision accuracy of a bioinformatics neural network system from the literature for protein structure prediction. We also describe an unsound assumption made by the authors of the neural network system, and propose a fuzzy hybrid solution, which eliminates the need for this assumption and can further enhance performance.
Keywords: Bioinformatics, Decision accuracy, Protein structure prediction, Hybrid system, Neural network, Fuzzy logic

1. Introduction

Protein is the machinery of life. It is required in all organisms for the structure, function, and regulation of the body's cells, tissues, and organs. Each protein has unique functions. For example, one type of protein, known as an enzyme, helps in our body's digestion system. The structure of a particular protein determines its function. The techniques to experimentally determine the 3D structure of proteins are complicated and time consuming. Determining a structure can take from one to several years.

A protein is formed by a chain of amino acids (hereafter known as a protein sequence). Over the years, many new proteins have been identified by large-scale genome sequencing projects. While the protein sequence of the new protein can be identified, the protein structure is often not known. As an attempt to narrow the gap between the number of known protein sequences and the number of experimentally determined protein structures, methods for protein structure prediction have been studied (Defay and Cohen, 1996; Fischer and Eisenberg, 1996; Flockner et. al, 1995; Lathrop and Smith, 1996).

L. Reznik et al. (eds.), *Soft Computing in Measurement and Information Acquisition*

In general, the protein structure prediction is performed by observing the protein sequence combined with our prior knowledge on a set of homologous proteins whose structure has been determined. At the time of writing, the prediction of a protein's three-dimensional structure from its amino acid chain (protein sequence) remains an unsolved problem. A review of the literature suggests that most of the research in this problem domain addresses the prediction of protein secondary structure.

Most of the work done in this problem domain attempts to predict a protein sequence to be one of the following: Helix (H), Extended (E) and Loop (L) (Zhang et. al, 1992). The problem can be viewed as a simple classification problem. Given a protein sequence, some algorithms are applied to classify the protein as Helix, Extended or Loop. Artificial neural networks are one of the predominant classifiers used in this problem domain (Qian and Sejnowski, 1988; Baldi et. al, 1999; Rost and Sander, 1993; Zhang et. al, 1992). Fuzzy logic and genetic algorithms have also been tried (Zhang et. al, ; Vivarelli et. al, 1995).

2. Research Goal

In this research, we explore the use of a fuzzy inference system (a.k.a. fuzzy system) to improve the protein secondary structure prediction accuracy of a successful neural network protein prediction system.

Since a fuzzy set allows for the degree of membership of an item in a set to be any real number between 0 and 1, this allows human observations, expressions and expertise to be modelled more closely. Once the fuzzy sets have been defined, it is possible to use them in constructing rules for fuzzy expert systems and in performing fuzzy inference. Fuzzy system can produce more accurate results based on the basic idea of the defuzzification. A defuzzification technique is used to calculate the conclusion by evaluating the degree of matches from the observation that triggered one or several rules in the model. This will lead to a better result by handling the fuzziness in the decision making. Thus, the fuzzy technique can improve the neural network prediction in certain cases.

Among the neural networks used for protein structure prediction, the PHD (Profile-Based Network from Heidelberg) (Rost and Sander, 1993) was one of the first to claim to have achieved an accuracy of more than 70%. For this reason, the PHD has been chosen to be our base system to implement our fuzzy improvements.

This research aims to improve the accuracy of the PHD prediction by using a fuzzy system. While the fuzzy system is used in conjunction with the PHD networks in this study, it is reasonable to generalise that the technique can be used with other neural networks for protein structure prediction. The main emphasis

here is the development of a technique that improves the performance of neural network-based protein structure prediction tools using fuzzy logic.

3. Materials and Methods

An overview of the PHD network is presented in the section below. Complete details can be obtained from the original paper (Rost and Sander, 1993).

3.1 The prediction process

The PHD protein secondary structure prediction is carried out based on the following steps:

1. For each input protein sequence, the SWISSPROT data bank is searched for protein sequences that are homologous to the input sequence. This is done by using a program called BLAST, which is based on a well-known fast alignment method. The output of the program is a list of protein sequences in the SWISSPROT data bank as well as their similarity (in percentages) to the input protein sequence.
2. The list of homologues identified by BLAST is then fed into a more sensitive profile-based multiple alignment program, known as MaxHom.
3. The multiple sequence alignment produced by MaxHom is then refined by applying a filter. Only sequences with a similarity to the input protein sequence higher than a threshold are selected for the prediction process.

3.2 The PHD neural network

Figure 1 summarizes the operation of the PHD Neural Networks. The PHD network is composed of multiple 3-layer neural networks (i.e. networks with 1 hidden layer). The entire architecture consists of 3 levels.

The first level is a neural network, known as the sequence-to-structure net (SQSCN). The network takes a protein sequence as input and predicts the protein secondary structure as output, thus the name sequence-to-structure. For the SQSCN, the input is given a window of 13 basic cells. Given a protein sequence, 13 amino acids within the sequence are examined at a time and the secondary structure of the amino acids in the middle (position 7) is predicted. The input to each basic cell in the SQSCN is the profile computed from the multiple sequence alignment (as discussed previously). The network produces 3 real numbers representing the probability of the prediction being Helix, Extended and Loop respectively (more on this later).

The structure-to-structure net (SCSCN) is found in the second level of the PHD Network. The input of this network is given by the output of the first level network (SQSCN). The underlying theory is that the secondary structure at position N of a protein is affected by the structures at nearby positions, such as N-2, N-1, N+1, N+2 ...etc. The input of SCSCN is given by a window of 17 basic cells, each cell takes in 3 real numbers (Helix, Extended and Loop) produced by the first level network (SQSCN). The output of SCSCN is again a set of 3 real numbers (the probabilities).

The SQSCN and SCSCN are trained separately, using the Backpropagation algorithm. There are altogether 2 SQSCNs and 9 SCSCNs produced by using slightly different types of approach and training data. For each input protein sequence, different outputs are computed by the differently trained neural networks.

At the third level of the PHD architecture, the outputs of all the second level networks (SCSCN) are averaged to product the final output. This level is called the jury level.

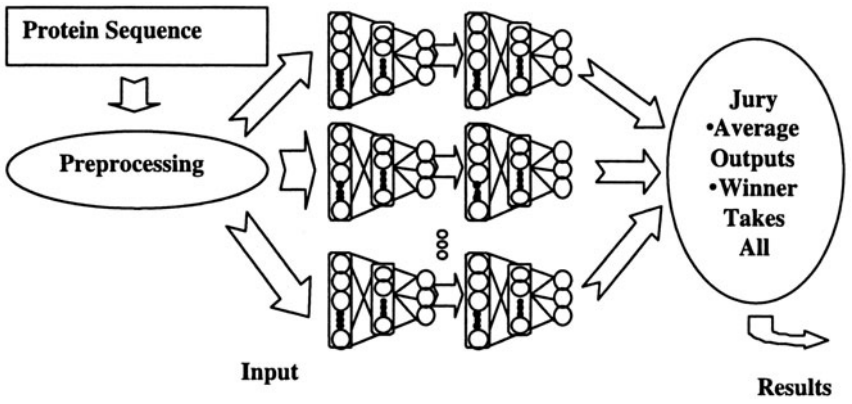

Figure 1: PHD Neural Networks in Operations

3.3 Research method

There is a major over-simplification in the PHD model, in that the network outputs are used as probabilities. The use of the outputs from neural networks which have been trained for classification as probabilities is not sound. The use of network outputs as probabilities is sound if the network has been trained using

probabilities. This is not the case in the PHD system, however this usage is understandable given the lack of availability of probabilistic data.

A further problem is the use of averaging in the jury layer which eliminates much of the dynamics of the predictions made by individual networks. Our methodology is to use a fuzzy system instead of the jury layer, and we also propose a fuzzy enhancement of the decision rules to improve the soundness of the decision making. This derives from the nature of fuzzy systems as possibilistic systems as opposed to probabilistic systems.

3.4 Fuzzy systems

Most fuzzy systems can be classified into three types (Jang et. al, 1997):

Mamdani style fuzzy system A fuzzy rule, with two input *(X,Y)* and one output *(Z)*, comes in the form *if X is **A** and Y is **B** then Z is **C***, where ***A, B*** and ***C*** are fuzzy sets. Since the overall output of the system is a fuzzy set, a defuzzication process is normally performed to compute a crisp value out of the resulting fuzzy output.

Sugeno style fuzzy system A fuzzy rule, with two input *(X,Y)* and one output *(Z)*, comes in the form *if X is **A** and Y is **B** then* $Z = pX+qY+r$, where ***A, B*** and ***C*** are fuzzy sets, *p, q, r* are parameters that are used in conjunction with the inputs to compute the output. No defuzzication is required in this type of inference system.

Tsukamoto style fuzzy system The fuzzy rules used in this type of fuzzy system are similar to those of Mamdani style with the exception that the fuzzy set in the consequent part is characterized by a monotonical membership function. As a result, the output of each rule is defined as a crisp value induced by the rule firing strength.

It is noted that the three types of fuzzy systems differ only in the consequent part.

3.5 Fuzzy system as jury level

Fuzzy sets allow human expertise and decisions to be modelled more closely, thus it is suggested that we can replace the third level of the PHD architecture, the jury level, by a fuzzy system. Each of the nine neural networks in PHD produces three outputs representing the probabilities of Helix, Extended and Loop respectively. Hence, the number of inputs to the jury level is $3 \times 9 = 27$.

Fuzzy systems are well known for their "curse of dimensionality". In general, the number of fuzzy rules grows exponentially with the number of input variables and number of fuzzy terms per input variable. Even for moderate number of inputs, the

number of rules involved in the fuzzy system can be very large. As a result, the operation and training of the fuzzy system become very slow.

As an effort to overcome this problem, we propose the use of multiple cooperative fuzzy systems as shown in figure 2. The task of secondary structure prediction is segregated to three independent fuzzy systems. The first fuzzy system, called the Helix fuzzy system, receives the Helix output from all the neural networks and output a value representing the probability that the final prediction is Helix. The second and third fuzzy systems perform similar tasks on the Extended and Loop predictions respectively. In this design, each fuzzy system deals with minimal amount of inputs. For simplicity, three out of the nine PHD neural networks are selected for the purpose of this research. That is, the fuzzy systems receive a total of 3 x 3 = 9 inputs.

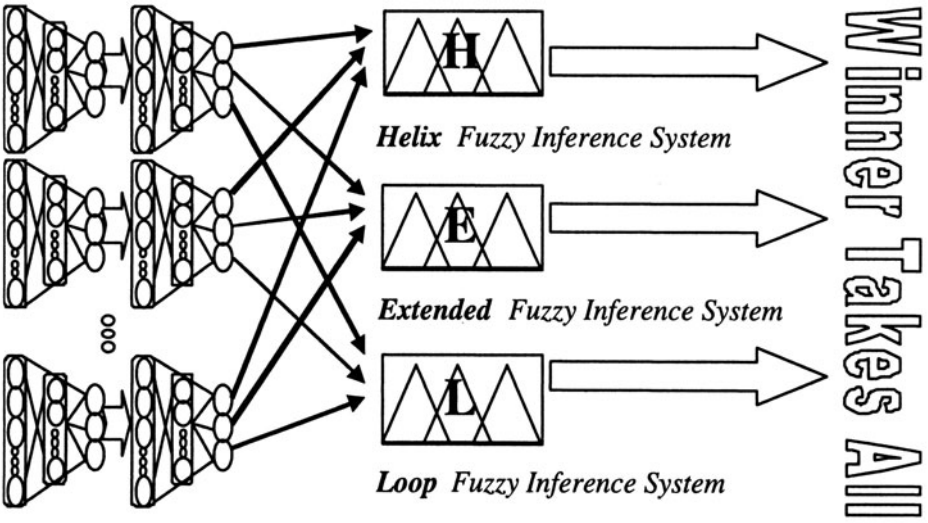

Figure 2: The use of Multiple Corporative

Pairs of input-output data are used to train the fuzzy systems. The data comes from the 126 protein sequences used in Rost and Sander (1993) are used to train the PHD neural networks. In the following sections, we would describe the training data and procedures used in this study.

3.6 Data

The input-output pairs come in the form:

$$(X_1^H, X_1^E, X_1^L, X_2^H, X_2^E, X_2^L, X_3^H, X_3^E, X_3^L; Y \in \{H, E, L\})$$

where X_i^T is the output of the i^{th} neural network representing the membership of class T, H = class Helix, E = class Extended, L = class Loop, and Y is the desired output. When training the three independent but corporative fuzzy systems, each input-output pair is split into three individual pairs:

$$\text{Helix Set: } (X_1^H, X_2^H, X_3^H; f(y, H))$$

$$\text{Loop Set: } (X_1^L, X_2^L, X_3^L; f(y, L))$$

$$\text{Extended Set: } (X_1^E, X_2^E, X_3^E; f(y, E))$$

where $f(s,t) = 100$ when $s = t$; 0 otherwise. In other words, we suppress the desired output of the individual pair for a fuzzy system if the desired output of the original pair is of the type which the fuzzy system predicts. By suppressing and depressing the proper individual pair, we allow the three fuzzy systems to learn the correct overall output in a corporative manner.

3.7 Adaptive-network-based fuzzy inference system (ANFIS)

The Adaptive-Network-Based Fuzzy Inference System (ANFIS) technique has been used in this research to generate the fuzzy rules from the training data (Jang, 1993). In this section, we outline the procedures involved in the technique.

ANFIS is a multi-layer feedforward network as shown in figure 3. Each node in the network is associated with a function. Some nodes have a set of parameters that are used in conjunction with the function to compute the output based on the input. Each node receives incoming signals from the previous layer and passes its output to the nodes in the next layer.

One point to take note is that unlike some of the common neural networks such as the multilayer perceptron, the links in an adaptive network do not have weights. They merely indicate the flow direction of signals between nodes.

It is interesting to note that ANFIS can mimic the function of a fuzzy system. By carefully designing the network structure, ANFIS can operate like any of the three types of fuzzy systems described in section 3.4. For the purpose of this study, only the Sugeno style fuzzy system will be discussed.

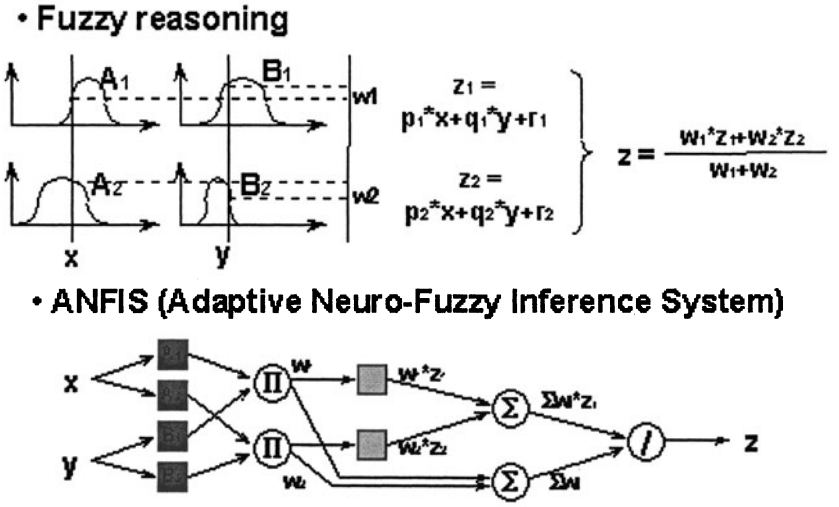

Figure 3: Sugeno Fuzzy System and ANFIS architecture

A typical ANFIS has five layers. Layer 1 is the membership layer. The functions associated with the nodes in this layer are membership functions used in fuzzy systems. The output of each node in this layer is the membership degree of the input. Common choices of the membership functions include the Triangular, Bell, Trapezoidal, and Gaussian function. Layer 2 is the rule layer. Each node output of this layer represents the firing strength of a rule. The function of each node can be any T-norm operators that performs fuzzy AND.

Layer 3 is the normalizing layer. The i^{th} node computes the ratio of the i^{th} rule's firing strength to the sum of all rules' firing strengths. Layer 4 is a layer that models the consequent part of a Sugeno fuzzy rule. We call it the consequent layer. It computes the output based on the normalized weights computed by the previous layer, system input (i.e. *x, y,* in the figure) as well as the three parameters (i.e. *p,q,r*) used in the consequent part of the Sugeno fuzzy rule. Layer 5 consists of only one node that sums up all the incoming signals and yields the output.

Hence, the network constructed operates similarly to the Sugeno fuzzy system. The adaptive structure enables the network to be trained using pairs of input-output data. Optimization techniques such as Backpropagation can be applied. However, it is well known that some of the common disadvantages of the gradient descent techniques are that they tend to be slow and become trapped in local minima. As a better alternative, the Hybrid Learning Algorithm has been proposed in Jang (1993), and is used in this study.

3.8 Training the fuzzy systems

The fuzzy rules of the Helix, Loop, and Extended fuzzy systems are learnt using ANFIS. The structure of the ANFIS used for this purpose is illustrated in figure 4.

There are three inputs (Helix, Extended, Loop) to the system. Three membership functions (Low, Medium, High) are used giving 3 x 3 = 9 nodes in the membership layer. The number of fuzzy rules involved in the fuzzy system is 3^3 = 27. This led to the 27 nodes in the rules and consequent layer. The lower node with the summation sign (Σ) in the figure represents the normalization factor (i.e. the sum of all firing strengths from the rule layer). This slight variation in design (as compared to figure 3) allows the normalization to be performed at the last layer. The upper summation sign node computes the weighted sum of the outputs from the previous layer. The individual node in the last layer divides the weighted sum by the normalization factor.

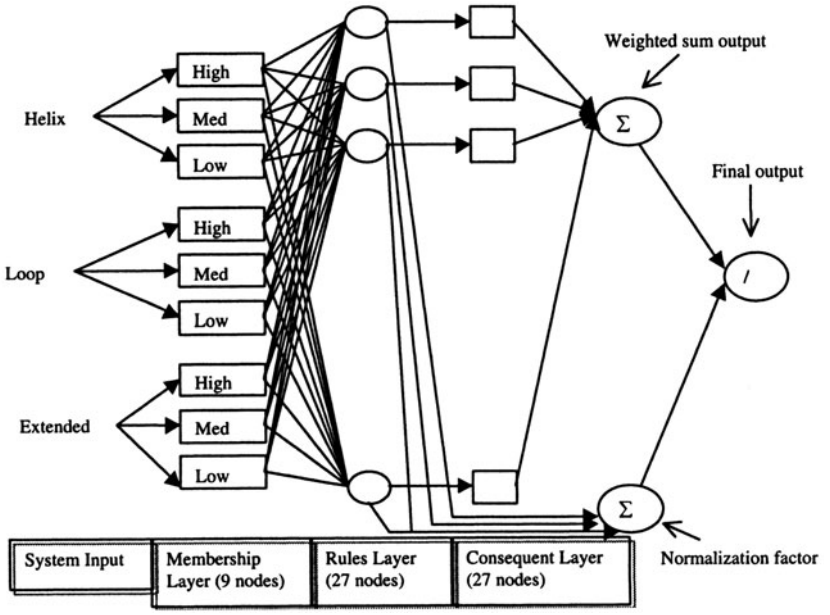

Figure 4: The ANFIS architecture of the Helix, Loop, and Extended fuzzy systems

The Hybrid Learning algorithm has been used to train the ANFIS for Helix, Extended and Loop. 4000 residues/entries are used for training. Training stops when one of the following occurs:

- The least mean squared (LMS) error is sufficiently small. In this case, the fuzzy system is considered completely trained and expected to achieve satisfactory performance.
- Over-fitting is sighted. To detect overfitting, a test set that is separate from the training set is prepared prior to the training. At the end of each epoch, the errors of both the training set as well as the test set are computed. In the beginning of the training, the errors of both sets are expected to decrease. The possibility of over-fitting can be observed when the error of the test set increases continuously.
- The decrease of LMS error is not significant over each epoch.

4. Results and Discussions

In general, accuracy can be computed:

$$\frac{Correctly\ predicted\ residues}{Total\ number\ of\ residues} \times 100\%$$

Each protein sequence contains, on average, a few hundred residues. Residues were randomly selected from different protein sequences as test data. Altogether 8000 residues that were distinct from the training set were used for testing. The test data was organized into 4 sets. Each set contains residues of several protein sequences. The test data were used to test both the original PHD network that operates with the jury layer as well as the improved version that operates with the three fuzzy systems.

The accuracy of the original PHD Network is measured to be 74.92%. The accuracy of the improved version is shown in table 1. The fuzzy system improves the performance of the jury layer, as expected. While the improvement is not large the result is useful, and indicates that there is benefit of the use of fuzzy systems in a hybrid manner with the existing neural network model. The next step is the extension to the individual networks.

Apart from the accuracy of the prediction system, other issues have also been investigated. It is observed that by segregating the jury task to three corporative fuzzy system, the training process has been sped up. A fuzzy system that accepts 9 inputs (from three neural networks), takes more than 40 minutes to complete one epoch. The decrease in least mean square error over each epoch is also

insignificant suggesting the possibility of slow convergence. This is in contrast with the use of three small fuzzy systems where each system requires less than 5 minutes to compete an epoch and converges, on average, in approximately 50 epochs. Table 2 shows the number of epochs required by each fuzzy system to complete training in this study. The benefits of using small and cooperative fuzzy systems over large fuzzy systems are obvious. By small fuzzy systems, we mean fuzzy systems that have small number of inputs. At the time of writing, designing fuzzy systems that requires large amount of inputs is still a difficult task. This is because fuzzy systems suffer from rules explosion. The use of multiple cooperative fuzzy systems could be a simple yet efficient technique to address part of this problem.

Table 1: Accuracy of PHD network with three fuzzy systems that replaces the jury layer

Test Set	Accuracy
1	*75.50%*
2	*74.75%*
3	*75.35%*
4	*74.45%*
Average	***75.01%***

Table 2: Epochs required by each fuzzy system to complete training

Fuzzy System	Epoch
Helix	*40*
Extended	*33*
Loop	*76*

Table 3: Several Fuzzy Rules of the Helix Fuzzy System

If (H is Low) and (L is Low) and (E is Medium) then O = -48.68H + 105.8L + 1.013E - 48.74
If (H is Low) and (L is Low) and (E is Low) then O = 89.43H + 8.439L + 1.14E + 5.863
If (H is Low) and (L is Medium) and (E is Low) then O = 71.92H + 8.105L - 49E – 375.8

AGENDA:
H = Helix
L = Loop
E = Extended
O = probability of the final prediction output to be Helix

There is another advantage to the use the fuzzy systems as compared to the jury layer. Fuzzy systems operate using fuzzy rules. Table 3 shows 3 out of the 27 fuzzy rules generated for the Helix fuzzy system. The use of fuzzy rules allows for user manipulation and interaction. It allows domain experts to adjust each fuzzy rule to improve the overall system performance. As a result, a system becomes more adaptable to changes over time. Domain experts can encode new knowledge into the system from time to time as well.

5. Conclusions

We proposed in this paper the use of individual fuzzy systems for each neural network's outputs in previous layers. We have noted earlier that neural network conclusions cannot safely be used as probabilities if the network has not been trained using probability information in the training set, but has been trained (as is usually the case) for classification using 0 and 1 (or 0.1 and 0.9) values.

The use of hybrid neural network and fuzzy systems can improve decision accuracy in a practical application in bioinformatics for protein structure prediction. This improvement derives from the more sophisticated defuzzification techniques available as compared to simple averaging. Using this technique, no probabilistic assumption is necessary, instead the network outputs are transformed into fuzzy membership values, which is expected to further enhance the accuracy of the overall system. We have also proposed the use of the fuzzy logic's possibilistic nature to replace the unsound probabilistic assumption often made for neural network conclusions. The latter is ongoing work which will be reported elsewhere.

As part of the study, we have also addressed another issue, the use of multiple cooperative small fuzzy systems, as opposed to large fuzzy systems. This can be beneficial. The amount of time that is required for training the three-input fuzzy systems is small.

The use of fuzzy systems in place of the jury layer of the neural networks also results in a more user-friendly system. Domain experts can encode knowledge into the fuzzy systems by adjusting the fuzzy rules. The fuzzy rules also help to explain the steps taken by the prediction system in producing a particular output.

References

Baldi, P., Brunak, S., Frasconi, P., Soda, G., and Pollastri, G. (1999): Exploiting the past and the future in protein secondary structure prediction. Bioinformatics, 15(11), 937-946.

Defay, T.R., and Cohen, F.E. (1996): Multiple sequence information for threading algorithms. J. Mol. Biol., 262, 314-323.

Fischer, D., and Eisenberg, D. (1996): Protein fold recognition using sequence-derived predictions. Protein Sci., 5, 947-955.

Flockner, H., Braxenthaler, M., Lackner, P., Jariz, M., Ortner, M., and Sippl, M.J., (1995): Progress in fold recognition. Proteins: Struct Funct. Genet, 23, 376-386.

Jang, J-S.R., (1993): ANFIS: Adaptive-Network-Based Fuzzy Inference System. IEEE Transaction on Systems, Man, and Cybernetics, 23(3).

Jang, J-S.R., Sun, C-T., and Mizutani, E. (1997): Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice Hall, US.

Lathrop, R.H., and Smith, T.F. (1996): Global Optimum protein threading with gapped alignment and empirical pair score functions. J. Mol. Biol., 255, 641-665.

Qian, Ning., and Sejnowski, T.J. (1988): Predicting the Secondary Structure of Globular Proteins Using Neural Network Models. J. Mol. Biol., 202, 865-884.

Rost, B., and Sander, C. (1993): Prediction of Protein Secondary Structure at Better than 70% Accuracy. J. Mol. Biol., 232, 584-599.

Vivarelli, F., Giusti, G., Villani, M., Campanini, R., Fariselli, P., Compiani, M. and Casadio, R. (1995): LGANN: a parallel system combining a local genetic algorithm and neural networks for the prediction of secondary structure of proteins. Bioinformatics, 11, 253-260.

Zhang, X., Mesirov, J., P., and Waltz, D.L. (1992): Hybrid System for protein secondary structure prediction. J. Mol. Biol., 225, 1049-1063.

Zhang, Chun-Ting, Chou, Kuo-Chen, and Maggiora, C.M.: Predicting protein structural class from amino acid composition: application of fuzzy clustering.