Pruning Neural Networks for Protein Secondary Structure Prediction

Sepideh Babaei, Seyyed A. Seyyedsalehi, and Amir Geranmayeh, *Member, IEEE*

***Abstract*—Secondary structure prediction is an effective approach in deducing the three dimensional structure and functions of proteins. Although the multilayer neural network is currently used for the prediction, appropriate determination of the network size is yet an important factor in improving the performance of the network. In this work, two systematic approaches for pruning the oversized multilayer perceptron neural networks (MLP-NN) are proposed to determine the optimum size of the hidden layer. Using the RS126 dataset in seven-fold cross-validation, the percentage accuracy of the prediction reaches to 75.38.**

I. Introduction

PROTEINS are large complex molecules that are made up by smaller subunits called amino acids. Chemical properties distinguishing the 20 standard amino acids cause the protein chains to fold up into specific three dimensional (3D) structures defining their particular functions in the cell. Secondary structure is the locally ordered structure created by hydrogen bonding within the protein backbone [1]. Experimental methods, such as X-ray crystallography and nuclear magnetic resonance spectroscopy that are used to determine the protein structure, are time consuming, labor expensive, and not applicable to all proteins [2]. These prohibitive costs may increase the gap between the number of known protein sequences and the number of known structures. Hence, the prediction of a protein structure from the amino acids sequence, initiated in late 1970s, is yet an important computational goal.

Computational methods usually perform the prediction of the 3D structure with an intermediate step of predicting the secondary structure. The early approaches were only based on the primary sequence information and they were able to predict three secondary structure types with an accuracy of less than 60% [3]. The next generation of the methods considered the information of neighboring amino acids through sliding-window computations [1]. These methods use pattern recognition and statistical characteristics based on Bayesian inference and decision rules, hidden Markov models, support vector machines, and neural networks [1] and they can achieve to a maximum accuracy of about 80% [4].

Nowadays, using neural networks is a promising approach in the secondary structure prediction [5]. Following the pioneering work of Qian and Sejnowski [3], many new computational techniques involving neural networks for the prediction of proteins secondary structure were introduced an average prediction accuracy that varies from 70 to 80%. In order to improve the prediction accuracy, several studies have been applied sophisticated network structures such as hierarchical [4], cascade [6], recurrent [7], bidirectional [8], and multiple experts networks [9]. Others combined additional structural information in the network input with the amino acid composition [10], interaction graphs [11], tertiary [12] and secondary [13] structure information, probabilities of the residues in the protein core or on the protein surface [14], multiple sequence alignment profiles [15], and position specific score matrices (PSSM) [16].

All in all, neural networks have some particular difficulties in defining the network architecture and structure as well as training algorithms. It is not a priori obvious what size of the network is the best. Small networks generalize properly, however, they might not be able to fully learn the data. On the other hand, the large ones learn slowly and prone to be so sensitive to the initial condition and learning parameters [17].

The main purpose of the present work is determining the optimum number of hidden layer nodes in the feed forward neural network, using pruning algorithms to eventually improve the accuracy of the prediction. The PSSM profiles of the RS126 dataset are applied as the input to sequence-to-structure network. The network with a larger size than that is required is initially trained. The optimum size of the hidden layer is then defined using two methods. In the first pruning method, the redundant nodes are removed based on the network training error. In the second method, the correlated neurons are unified using Sietsma and Dow approach. After this stage, the network output is fed the structure-to-structure network as proposed in the profile network from Heidelberg (PHD) method [15].

The organization of the paper is as follows. In Section II, the structural change and development of the neural network is reviewed. The employed dataset, experimental preparation, and evaluation method are described in Section III. Section IV explains the proposed pruning algorithms for the neural network classifier architecture. The experimental results are discussed in Section V, and finally, the

Manuscript received July 5, 2008.

S. Babaei is with the Biomedical Engineering Department of Amirkabir University of Technology (Tehran Polytechnic), P.O. Box: 15875-4413, Tehran, Iran (corresponding author to provide phone: +98-21-64562353; fax: +98-21-88278817; e-mail: sepeedeh.babai@gmail.com).

S. A. Seyyedsalehi is with the Biomedical Engineering Department of Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran (email: ssalehi@aut.ac.ir).

A. Geranmayeh is with the Department of Electrical Engineering and Information Technology, Darmstadt University of Technology, 64289 Darmstadt, Germany (e-mail: geranmayeh@temf.tu-darmstadt.de).

conclusions are drawn in section VI.

II. Structural Change and Development in Neural Networks

In general, the number of hidden units, or equivalently weights, that are needed to produce a reasonable approximation to the data is not clear a priori. Guessing an appropriate number is the most usual answer but, of course, not necessarily the best. Another common solution is to try out several network sizes and select the most promising one [17]. Neither of these methods is very principled. Meanwhile, neurons meiosis, immigration, death and adhere may occur in developmental neurobiology that are known as progressive changes [18]. In this regard, the dynamical connectionist artificial neural networks are developed based on processor units and connections removing, pruning and growing algorithms [19]. In the growing networks such as the meiosis networks, generally, a very small size network is trained, and iteratively, the complementary units and weights are added. This is accomplished by using the neurons weights variances to create hidden nodes dynamically, the back propagated error to increase hidden neurons, and network training error to modify the neurons weights [17].

An alternative technique to the growing networks is to start with a relatively large network and then remove weights to fulfill optimal network architecture. Exploring the less important weights is a difficult issue for which several heuristic approaches have been proposed [17]-[19]. Among them, skeletonization pruning method nullifies every weight and examines the resulting changes in the network training error individually [19]. This technique stems from the observation of the functional importance variation of the units after training. The results are monitored during the removal of the unit and its connections to explore whether a unit is functionally important. The relevance of a unit can then be defined in terms of the difference in the network training error. If the error discrepancy is larger than a priori set threshold then the unit functionality is useful and it is retained, otherwise, the effect of unit is negligible and it is removed.

Considering the omission types of redundant nodes, the pruning methods are categorized into two types. Those which evaluate the sensibility of the error function to truncate the elements with less influence are known as the methods of sensibility. They modify the network once the training is done, the sensibility is calculated, and based on the value of the weights, the redundant nodes are cut away. The other pruning type performs the removing after adding a penalization term to minimize the function. This accommodates the network for choosing efficient solutions [17].

In addition to the mentioned processes, there exist techniques with particular methodologies. Sietsma and Dow [20] described an interactive method on which a designer inspects a trained network and decides which nodes to eliminate. The idea is based on removing the noncontributing nodes. If the output of a unit is approximately constant for the training patterns, then it is acting like an additional bias to all nodes. As a result, unification of nodes with highly correlated outputs during the training can serve as an alternative approach to prune the network. In this paper, the Sietsma removing algorithm is applied to reduce the network size as well as, skeletonization pruning method.

III. Data Preparation

A. Secondary Structure Assignment

The assignment of the protein secondary structure can be performed by three programs, namely the DSSP, STRIDE and DEFINE [21]. In this work, the define secondary structure of proteins (DSSP) assignment is adopted. According to this method the secondary structure of each residue classifies into 8 classes, namely H (α-helix), G (310-helix), I (π-helix), B (isolated β-bridge), E (extended β-strand), T (hydrogen bonded turn), S (bend), and C (not HBEGIT or S). The prediction methods are normally assessed for only 3 standard classes associated with α-helix (H), β-strands (E), and coils (C). Hence, the 8 classes are reduced to 3 [1]. There are four main methods to perform the reduction process:

H {H, G}, E {E}, C {S, T, B, I, C} (I)
H {H}, E {E}, C {G, S, T, B, I, C} (II)
H {H, G, I}, E {E, B}, C {S, T, C} (III)
H {H, G}, E {E, B}, C {S, T, I, C} (IV)

Here, the method (IV), so-called the critical assessment of techniques for protein structure prediction (CASP), is adapted. It is considered as the strictest criterion and usually results in lower prediction accuracy than the other methods [21]. In order to encode the secondary structure classes for the classifier, the three units are assumed as binary values according to the following allocation

H=[1,0,0] E=[0,1,0] C=[0,0,1].

B. RS126 Dataset

The RS126 main dataset are applied to develop and test the predictor. It contains 126 non-homologous globular proteins according to the definition given by Rost and Sandar [15]. They applied percentage identity to measure the homology and defined non-homologous to signify that no two proteins in the dataset share more than 25% sequence identity over a length of more than 80% residues [15]. Numerous protein secondary structure prediction methods are being developed and tested on the RS126 dataset. The dataset comprises 24395 amino acids with secondary structure extent 32% α–helix, 21% β–strand and 47% coil.

C. Evaluation Method

With seven-fold cross-validation approximately 1/7 of the proteins in the dataset are left out for testing and the rest is

used for training. This procedure is fulfilled cyclically seven times and the prediction result is a mean over seven different testing sets. In order to avoid the selection of extremely biased partitions that may give inauthentic prediction accuracy, the RS126 set is partitioned into seven subsets with equal size and similar content from each type of the secondary structures. Several different random partitions of the RS126 set are tested. The partition that distributes the three secondary structure types (H, E, and C) most evenly is selected finally.

The percentage of residues predicted correctly in the conformational state k is given by

$$Q_k = \frac{n_k}{N_k} \times 100 \quad (1)$$

where k represents H, E and C regions in the native protein structure as determined experimentally, n_k is the number of correctly predicted residues in the state k, and N_k is the total number of residues in the conformational state k in the test set. The percentage of the total residues correctly identified in the three classes is obtained by

$$Q_3 = \frac{N_H + N_E + N_C}{N_T} \times 100 \quad (2)$$

where N_T is the total number of residues in the proteins set and N_H, N_E, N_C indicate the number of correctly classified amino acids belonging to the corresponding classes.

D. Position-Specific Score Matrices (PSSM)

Prediction based on a multiple alignment profile of protein sequences instead of a single sequence has long been recognized as a way to improve the prediction accuracy [15], [16]. There are two kinds of alignment profiles: the multiple sequence alignment profiles (MSAP) and the position-specific score matrices (PSSM). In this paper, the PSSM on the RS126 set are utilized. The profile matrix has $20 \times L$ elements, where L is the length of the target sequence, and each element represents the occurrence frequencies of the i-th amino acid in the j-th position, as expounded in [16]. The elements are typically in the range ± 7 and normalized between 0 and 1 using the standard logistic function

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

where x is the raw profile matrix value.

IV. Classifier Architecture

A. Sequence-to-Structure Network

The supervised classifier is based on a multilayer perceptron (MLP) network with only one hidden layer. It associates the primary structure with the secondary one, only considering a PSSM profile of single amino acid and its neighbors. Through the interactions between the constituent amino acids along a protein chain, the residue neighbors affect the relevant secondary structure. Thus, using the sliding window on the amino acids sequence and feeding the classifier are the most common solution to improve the prediction accuracy. There are no specific rules to specify the length of the window. It has been shown the best choice can be found among the odd numbers between 9 and 51 [5].

After selecting the neighboring amino acids, the pattern vector for the i-th residue is built. It is necessary to identify the rows of the PSSM matrix of neighboring amino acids and subsequently to lexicographically concatenate them in the i-th residue pattern vector. According to the RS126 dataset the optimum length of window is 13 residues [15], [22]. Hence, the network training parameters are adjusted similar to Rost and Sander method [15]. As a result, the pattern vector is made up of $N_X = 13 \times 20 = 260$ elements. Indeed, 13 rows of the PSSM matrix containing 20 elements are concatenated. The window slides along the protein chain and the patterns are computed for all of the amino acids of the sequence. Null rows are considered for the PSSM as long as the window is on the head or end of the chain.

The network output vector representing the 3 secondary structure classes is obtained by

$$Z_t^i = f\left(\sum_{h=1}^{N_Y} W_{Y-Z}^{ih}\left[f\left(\sum_{j=1}^{N_X} W_{X-Y}^{hj} X_t^j + \theta_1\right)\right] + \theta_2\right) \quad (4)$$

where X is the network input vector with length N_X and W_{X-Y}^{hj} denotes the weight associated with the j-th unit of X to the h-th unit of the hidden layer vector Y with length N_Y. The coefficient W_{Y-Z}^{ih} is the connection weight between the h-th unit of hidden layer and the i-th unit of the network output. The neurons activation function f is logistic. Bias weights θ_1 and θ_2 are added to the input and hidden layers, respectively. The desired output is the central residue, namely the seventh entry of the input window, class. The secondary structure at position t is predicted to be C_i if $Z_t^i > Z_t^j$ for all $j \neq i$, where $C_i \in \{H, E, C\}$.

The number of neurons of the hidden layer is an important issue in gaining the optimum performance of the classifier. A systematic variation of the hidden nodes number has not yet been studied. According to the pervious studies, some authors tried out several network sizes and selected the size that gives the desired accuracy [23]. The number of exploited hidden nodes is varying in the range of 30 to 80 [15], [22], [23]. Thus, an oversized network with $N_Y = 100$ nodes in the hidden layer is trained on the seven training groups of the dataset.

The back propagation algorithm is applied to train the fully-connected feed-forward network, using both a constant learning rate and a momentum term. Training is terminated when either the error reduces to less than a priori set threshold or the training epochs reach an upper limit. The

threshold value is set 0.1 and the maximum number of training epochs is adjusted to 2000 epochs. At each training epoch, the samples of the training set are fed in randomly changing orders. The training error for the first group of the dataset is illustrated in Fig. 1. Specifically, the following settings are used:

- A constant learning rate is 0.05.
- A momentum parameter is adjusted 0.5.
- The activation functions are logistic function sigmoid in [0, 1].
- Every parameter is initialized with small random values within [-0.1, 0.1] interval.

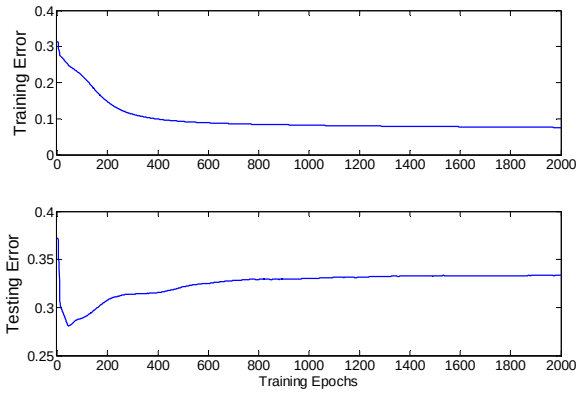

Fig. 1. Training and test errors of the oversized network for the first group of the input data (A). The error on the test set starts increasing after some epochs since the network is overtrained.

B. Pruning Algorithms

The protein secondary structure prediction is greatly prone to overtraining. Fig. 1 demonstrates the overtraining during the oversized network training. In the early stage of training the error on both training and test sets tends to decrease as the network is able to generalize from the examples to the underlying classes. However, the error on the test set begins to increase after some epochs, whereas the network starts to adapt artifacts in the training data. The hidden layer size in the MLP networks plays an important role in preventing the overtraining. Hence, determining the number of hidden nodes with the systematic approaches is proposed.

The two pruning methods are applied to accomplish the best size of the network. The node removing in the first pruning algorithm (skeletonization) is based on the network relevance to the hidden neurons. The effect of any node on the training error is considered via evaluation of the trained network with elimination of the relevant node. The total error on the training data is calculated by

$$E = \sum_{t=1}^{P} \frac{1}{2}(Z_t - D_t)^2 \qquad (5)$$

where Z_t is the actual output of the i-th residue in training data, D_t is the target output vector (the associated class of Z_t), and P is the number of training patterns. The node is retained if the error increases more than the set threshold 0.01. As a result, the weights with less relevance are truncated and the hidden layer is modified. The algorithms iterates 100 times for inspecting all of the hidden nodes.

In the second pruning technique, the noncontributing nodes in hidden layer are removed according to Sietsma and Dow method [20]. The unit with approximately constant output or a mimic output of another unit across the training set can be omitted. If the output of a unit is constant, then it is acting like an additional bias. Therefore, the average output of the unit is added to the bias node after removing. The bias weight modification moves the hyper planes place in the feature space and changes the classification result. Whenever, this approach is applied to prune the network, the classification improvement is insignificant. Thus, the alternative method that is unification of units with identical outputs is employed.

Two nodes are combined whenever the distance between the corresponding outputs is less than the constant threshold which is assigned 0.05. The hidden layer establishes two connections with the input layer ($W_{X\text{-}Y}$) as well as the output layer ($W_{Y\text{-}Z}$). Thus, the associated neurons of the approximated analogous nodes are correlated and considered as a single neuron during the unification. The final weight of obtained neuron is substituted by average of the weights. The examination is repeated during the pruning to gain reduction in the network size.

The other main advantage of the neural network pruning is defining the optimum size of the network. Hence, the average of obtained hidden layer sizes on the 7 groups of training data from the two employed pruning methods are applied for defining the proper classifiers. The secondary structure prediction is then tested using two fully-connected feed-forward neural networks with two differences sizes. The sliding window with length of 13 on amino acids sequences is fed into the networks. The networks training parameters such as constant learning rate and momentum term are adjusted similar to the prior networks. At each training epoch, the training patterns are fed into the network in randomly changing orders and the training is terminated when either the training epochs reach 1000 or the error reduces to less than 0.2.

C. Structure-to-Structure Network

The first network regards the dependency of a residue and its secondary structure using ±6 residue neighbors of central amino acid in the 13 unit wide window. The consecutive secondary structures are correlated, e.g., α-helix consisting of at least 3 consecutive patterns [1]. According to the long range correlation between the types of secondary structures along the protein chain, a second level network is exploited to take into account the neighboring effects. The input of the networks is fed using a sliding window with length 17. In other words, the network acts as a filter that processes the output of the first networks to enhance the prediction accuracy [15]. Therefore, the one hidden layer network is

trained by the actual first network output Z_t in (4), together with the 8 neighboring vectors of Z_t as the input and D_t as the desired output. The constant learning rate and momentum term are similar to those of the first network. The network is trained using back propagation algorithm. The training is terminated at the 1000th epoch. Fig. 2 presents the training error of the network for the first group of the training dataset when the network is fed by the output of the oversized network.

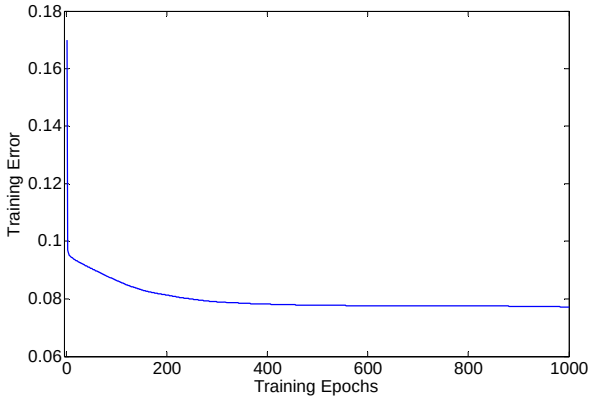

Fig. 2. Training error of the structure-to-structure network for the first group of the input data (A).

V. Results and Discussion

The secondary structures of the RS126 dataset are predicted through 6 diverse neural networks using 7-fold cross-validation. The one hidden layer feed-forward neural network with 100 nodes in the hidden layer is considered as the oversized network (OS-Net). The network maps the amino acids sequences to the secondary structures using back-propagation training algorithms. Structure-to-structure network (SS-Net) is applied to filter the output of the first networks. The accuracy of prediction for three secondary structure classes (Q_H, Q_E, Q_C) and the total accuracy (Q_3), referring to (1) and (2), are given in Table I. As expected, the accuracy of prediction indicated by Q_3 improves using the SS-Net by considering the interaction between the secondary structures. Particularly, the prediction accuracy of the β-strands type (Q_E) increases considerably, since the long range interaction between secondary structures is more effective in β-strands than α-helix and coils.

TABLE I
Prediction Accuracy of the Oversized and Structure-to-Structure Net

Type of Net	Q_H	Q_E	Q_C	Q_3
OS-Net	74.58	65.51	72.64	71.09
SS-Net	74.83	66.53	72.76	71.82

Regarding the effect of the each node on the training error, the OS-Net is pruned with skeletonization pruning method and the units with insignificant influence are removed. The algorithm is performed on all networks trained by the 7 groups of the dataset {A, B, C, D, E, F, G}. The achieved size of the hidden layer for the data groups and the prediction accuracy after removing are presented in Table II. The OS-Net is also pruned using the second pruning algorithm based on the Sietsma and Dow method. The correlated weights are adhered and the optimum size of the networks is attained. The numbers of hidden nodes for all training datasets and the relevant Q_3 have been tabulated in Table III.

TABLE II
First Pruning Method for the 7 Groups of the Dataset

Data Group	A	B	C	D	E	F	G
Num. of Nodes	78	85	83	79	85	77	81
Q3	69.87	78.57	78.06	72.38	77.37	73.93	69.94

TABLE III
Second Pruning Method for the 7 Groups of the Dataset

Data Group	A	B	C	D	E	F	G
Num. of Nodes	70	77	74	71	75	71	73
Q3	68.93	78.13	78.67	71.94	77.86	73.26	69.2

During network training, the weights are modified based on the back-propagation rule in order to decrease the error on training patterns. When the network is tested on different dataset, it generalizes the training data to the test data. The network with excessive nodes in hidden layer may learn more details from the training data and the training error decreases sufficiently, however, the test error starts increasing after some training epochs. The network is not able to generalize and the overtraining is accrued. However, by pruning the hidden layer the overtraining is avoided. The reported prediction accuracies in Table IV demonstrate the consequence of removing less important nodes in increasing the accuracy.

TABLE IV
Prediction Accuracy Obtained Using the Pruning Algorithms

Pruning	Ave. Nodes	Q_H	Q_E	Q_C	Q3	Q_3 SS-Net
Method1	**81**	75.89	68.75	73.71	74.3	**75.24**
Method2	73	75.52	68.89	73.1	73.99	74.76

The SS-Net improves the accuracy whenever it is applied after two pruning methods. Using the first pruning method the highest Q_3 = 74.3% and 75.24% are attained respectively from the pruned sequence-to-structure and the structure-to-structure networks as shown in Fig. 3.

The optimum size of the hidden layer is defined by averaging on the obtained number of hidden nodes for the 7 groups of the dataset. Finally, 81 and 73 hidden layer nodes are obtained respectively for the first and the second pruning methods. The two feed-forward neural networks with one hidden layer are applied for classification. The first network (Net1) with 81 and the second network (Net2) with 73 nodes in the hidden layer are trained by the 7 groups of the dataset.

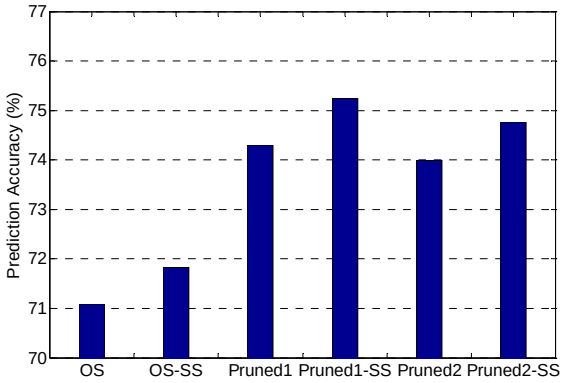

Fig. 3. The prediction accuracy of the six proposed networks; the structure-to-structure network for the first pruning method achieves the highest classification performance.

The outputs of the networks are then optimized using the SS-Net with the prior settings. The prediction accuracy of three secondary structure types (Q_H, Q_E, Q_C) and the total accuracy (Q_3) are given in Table V. Considering the final accuracy of the designed network (75.38%) the results are comparable with Q_3 of the well-known predictors such as those reported by Jpred (74.8%) [24] and SSpro (78.1%) [25].

TABLE V
PREDICTION ACCURACY OF USING THE NETWORKS WITH 73 AND 81 NODES IN HIDDEN LAYER AND THE STRUCTURE-TO-STRUCTURE NET

Type of Net	Q_H	Q_E	Q_C	Q_3
Net1	71.04	72.84	78.15	74.25
SS-Net1	71.93	74.47	78.82	**75.38**
Net2	71.19	72.76	77.68	73.84
SS-Net2	72.23	74.19	78.39	75.06

VI. CONCLUSION

Pruning algorithms were proposed to prevent the overtraining problem in the protein secondary structure classification by the neural networks approach. Using the pruning methods, the optimum size of the neural network is defined in a systematic way. The applied removing methods, particularly the skeletonization method based on the effect of every node on the training error, improve the classifier performance in practice. The accuracy of prediction is increased using the structure-to-structure network. The experimental results reveal that the prediction accuracy reaches to a level comparable with the other methods.

REFERENCES

[1] P. Baldi and S. Brunak, *Bioinformatics: The machine learning approach*, Cambridge, MA: MIT Press, Second Edt., 2001, ch. 1, 4, 6.

[2] Z. Aydin and Y. Altunbasak, "A signal processing application in Genomic research: Protein secondary structure prediction," *IEEE Signal Processing Mag*., vol. 23, no. 4, pp. 128-131, 2006.

[3] N. Qian and T. Sejnowski, "Predicting the secondary structure of globular proteins using neural network models," *J. Mol. Biol.*, vol. 202, pp. 865-884, 1988.

[4] L. Han, J. Cui1, H. Lin, Z. Ji, Z. Cao,Y. Li, and Y. Chen, "Recent progresses in the application of machine learning approach for predicting protein functional class independent of sequence similarity," *Proteomics*, vol. 6, pp. 4023–4037, 2006.

[5] B. Rost, "Neural networks predict protein structure: hype or hit?," *Artificial Intelligence and Heuristic Methods in Bioinformatics*, 2003.

[6] J. Chen and N. Chaudhari, "Cascaded bidirectional recurrent neural networks for protein secondary structure prediction," *IEEE Trans. Computational Biology and Bioinformatics*, vol. 4, no. 4, 2007.

[7] A. Ceronia, P. Frasconia, and G. Pollastri, "Learning protein secondary structure from sequential and relational data," *Neural Networks*, vol. 18, pp. 1029–1039, 2005.

[8] J. Chen and N. Chaudhari, "Bidirectional segmented-memory recurrent neural network for protein secondary structure prediction," *Soft Computing*, vol. 10, pp. 315–324, 2006.

[9] S. Sivan, O. Filo, and H. Siegelmann, "Application of expert networks for predicting proteins secondary structure," *Biomolecular Engineering*, vol. 24, pp. 237–243, 2007.

[10] J. Ruan, K. Wang, J. Yang, L. Kurgan, and K. Cios, "Highly accurate and consistent method for prediction of helix and strand content from primary protein sequences," *Artificial Intelligence in Medicine*, vol. 35, pp. 19-35, 2005.

[11] J. Cheng and P. Baldi, "Three-stage prediction of protein β-sheets by neural networks, alignments and graph algorithms," *Bioinformatics*, vol. 21 suppl. 1, pp. 75–84, 2005.

[12] G. Zhang, D. Huang, Y. Zhu, and Y. Li, "Improving protein secondary structure prediction by using the residue conformational classes," *Pattern Recognition Letters*, vol. 26, pp. 2346–2352, 2005.

[13] S. Costantini, G. Colonna, and A. Facchiano, "Amino acid propensities for secondary structures are influenced by the protein structural class," *Biochemical and Biophysical Research Communications*, vol. 342, pp. 441–451, 2006.

[14] G. Zhang, D. Huang, and H. Qiang Wang, "Protein secondary structure prediction based on the amino acids conformational and neural network technique," *IEEE ICASSP*, Canada, Montreal, May 2004.

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

[16] D. Jones, "Protein secondary structure prediction based on position-specific scoring matrices," *J. Mol. Biol*., vol. 292, 1999.

[17] R. Reed, "Pruning algorithms- a survey," *IEEE Trans. Neural Networks*, vol. 4, no. 7, pp. 740-747, 1993.

[18] D. Pures, L. White, and R. Riddle, "Is neural development Darwinian?," *Trends in neurosciences*, vol. 19, no. 11, pp. 460-464, 1996.

[19] P. Quinlan, "Structural change and development in real and artificial neural networks," *Neural Networks*, vol. 11, no. 4, pp. 577-599, 1998.

[20] J. Sietsma and R. Dow, "Creating artificial neural networks that generalize," *Neural Networks*, vol. 4, pp. 67-79, 1991.

[21] J. Cuff and G. Barton, "Evaluation and improvement of multiple sequence methods for protein secondary structure prediction," *PROTEINS: Structure, Function, and Genetics*, vol. 34, pp. 508–519, 1999.

[22] M. Mirto, M. Cafaro, S. Luigi Fiore, D. Tartarini, and G. Aloisio, "A grid-enabled protein secondary structure predictor," *IEEE Trans. on Nanobioscience*, vol. 6, no. 2, 2007.

[23] K. Guimaraes, J. Melo, and G. Cavalcanti, "Combining few neural networks for effective secondary structure prediction," in *Proc. 3rd IEEE Symp. on BioInformatics and BioEngineering*, July 2003.

[24] C. Cole, J. Barber, and G. Barton, "The Jpred 3 secondary structure prediction server," *Nucleic Acids Research*, vol. 36, pp. 197-201, 2008.

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

[26] H. Shen and K. Chou, "Ensemble classifier for protein fold pattern recognition," *Bioinformatics*, vol. 22, suppl. 1, pp. 1717–1722, 2006.