Journal of Theoretics *Vol.5-1*

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

Nazar M. Zaki, S. Deris, and S. N. V. Arjunan
Department of Software Engineering,
Faculty of Computer Science & Information Technology,
University Technology Malaysia, Skudai 81300, Johor, Malaysia.
nazar@siswa.utm.my

R. M. Illias
Department of Bioprocess Engineering,
Faculty of Chemical and Natural Recourses Engineering,
University Technology Malaysia, Skudai 81300, Johor, Malaysia

Abstract: Protein sequences classification is an important problem in molecular biology, and it has long been a goal for scientists and researchers. This paper describes an approach to data-driven discovery of sequence motif-based models using neural network classifier based on Dempster-Shafer Theory for assigning protein sequences to functional families. A training set of sequences with unknown functional family is used to capture regularities that are sufficient to assign the sequences to their respective families.

A new adaptive pattern classifier based on neural network and Dempster–Shafer theory of evidence developed by Thierry Denoux[1] is presented. This method uses reference patterns as items of evidence regarding the class membership of each input pattern under consideration. This evidence is represented by Basic Belief Assignments (BBA) and pooled using the Dempster's rule of combination. This procedure can be implemented in a multilayer neural network with specific architecture consisting of one input layer, two hidden layers and one output layer. The weight vector, the receptive field and the class membership of each prototype are determined by minimizing the mean squared differences between the classifier outputs and target values.

Keywords: functional family, protein sequence, neural networks, Dempster-Shafer theory.

1. Introduction

The last few years have witnessed consistent improvements in information retrieval, classification and analysis of the proteins and DNA sequences.[2] Early work on protein pattern recognition[3] suggested that subsequences of amino acids may be conserved in a protein family.

Using this observation, many approaches have been taken to discover these conserved regions and using them for protein function prediction. Currently, using these tools several databases have been developed to store these motifs. Examples of such databases include: Prosite[4], Pfam[5], and Prints[6] databases. One method for protein prediction is to query these databases to see if a protein contains any motifs in the database. The database then returns a function corresponding to any motif found in the protein. Sometimes a protein can contain several motifs. So an alternative approach would be to look at the presence or absence of an arbitrary number of combinations of motifs to determine protein function. Unlike approaches that try to classify protein sequences based on detecting a single motif within the sequence. This research describes an approach to data-driven automated discovery for assigning protein sequences to functional families based on the motif composition of the sequences.

2. Methodology

The basic computational problem we seek to address is that, given a database or training set of amino acid sequences that code for proteins with known function, our goal is to induce a classifier that would be able to assign novel protein sequences to one of the protein families represented in the training set. The basic approach is illustrated in Figure 1.

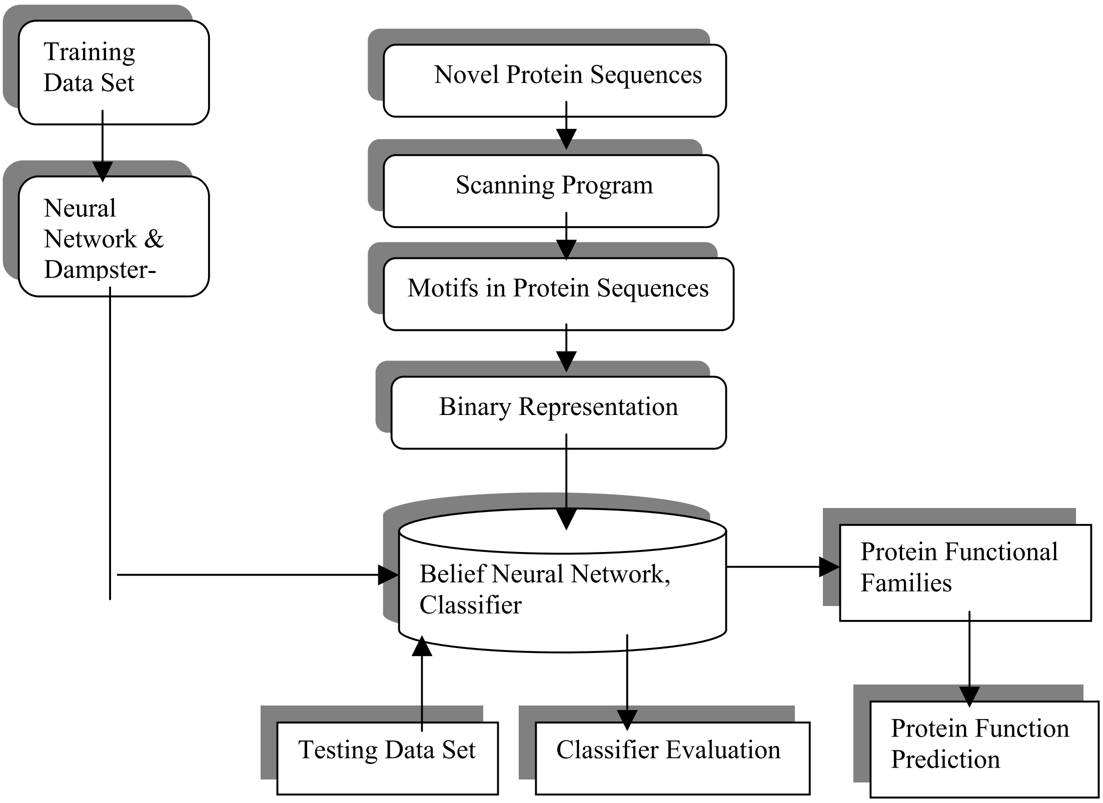
Training Data Set
Neural Network & Dampster-
Novel Protein Sequences
Scanning Program
Motifs in Protein Sequences
Binary Representation
Belief Neural Network, Classifier
Protein Functional Families
Testing Data Set
Classifier Evaluation
Protein Function Prediction

Figure 1: *Protein sequences with known functions are divided into a training dataset and a testing dataset, respectively. Belief Neural Network algorithm is used to build a classifier using the training dataset. Classification accuracy of the belief neural network is determined using the testing dataset. Finally, the classifier will be used to assign novel protein sequences to known functional families based on the proteins' motif compositions.*

2.1 Data Preparation

The Prosite database contains over 1100 entries. Each entry describes a function shared by some proteins. In this paper, one Prosite documentation entry corresponds to a protein class. The protein classes considered in this study are shown in Table 1, for clarity of presentation, the Prosite documentation ID, i.e., the PDOCxxxxx number, was used to represent that class. Similarly, the Prosite access number, i.e., the PSxxxxx number, was used to represent that motif pattern or profile. In the Prosite database, a protein motif can be a regular expression (defined over the 20 amino acid alphabet), called pattern, or a weighted matrix (built on alignment of multiple protein sequences), called profile.

Table 1: *The training sets of the protein classes considered in this study.*

Accession Number	Family Document	Class
PS00847, PS50051	PDOC 00662	Class of DNA or RNA associated proteins
PS 00066, PS 00318 PS 01192, PS 50065	PDOC 00064	Class of oxidoreductases)
PS 00862	PDOC 00670	Class of transverses
PS 50007, PS 50008	PDOC 50007	Class of hydrolase's
PS 00170, PS 50072	PDOC 00154	Class of isomerase's
PS 00411, PS 50067	PDOC 00343	Class of structural proteins
PS 00652, PS 50050	PDOC 00561	Class of receptors
PS 00251, PS 50049	PDOC 00224	Class of cytokines and growth factors
PS 00299, PS 50053	PDOC 00271	Class included in the catch-all "Others" category

Each protein class can be characterized by one or more characteristic motif patterns and/or profiles. For example, class PDOC00670 has two characteristic motifs, PS00856 a pattern and PS50052 a profile. Protein sequences containing any of the characteristic motifs of a functional class were collected and labeled as belonging to that class. Each collected protein was then processed by the profileScan program to determine its motif composition. Only the motifs that were identified as *significant* matches by profileScan were chosen. This analysis identified additional motifs in the sequences besides the ones designated as the characteristic motifs for the family associated with each sequence. Thus, each protein sequence was represented using binary attributes with each attribute denoting the presence or absence of the corresponding motif in the sequence. The presence of a known motif is presented by 1 or 0 otherwise (see Figure 1). Experiments in this paper were carried out mainly with proteins in this data set.

2.3 Denoeux Beliefs Neural Network Pattern Classifiers

An adaptive version of this evidence-theoretic classification rule is proposed. In this approach, computing distances to a limited number of prototypes, resulting in faster classification and lower storage requirements, makes the assignment of a pattern to a class. Based on these distances and on the degree of membership of prototypes to each class, BBAs is computed and combined using Dempster's rule. This rule can be implemented in a multilayer neural network with specific architecture consisting of one input layer, two hidden layers and one output layer. The weight vector, the receptive field and the class membership of each prototype are determined by minimizing the mean squared differences between the classifier outputs and target values.

Denoeux Belief Neural Network DBNN is a classifier based on the Dempster-Shafer theory of evidence. It uses training patterns as items of evidence for the class membership of each test pattern under consideration Figure 2. The evidence is represented by basic belief assignments (BBAs) and combined using the Dempster's rule. It is implemented in a multilayer neural network with one input layer, two hidden layers (activation and combination layers) and one output layer.

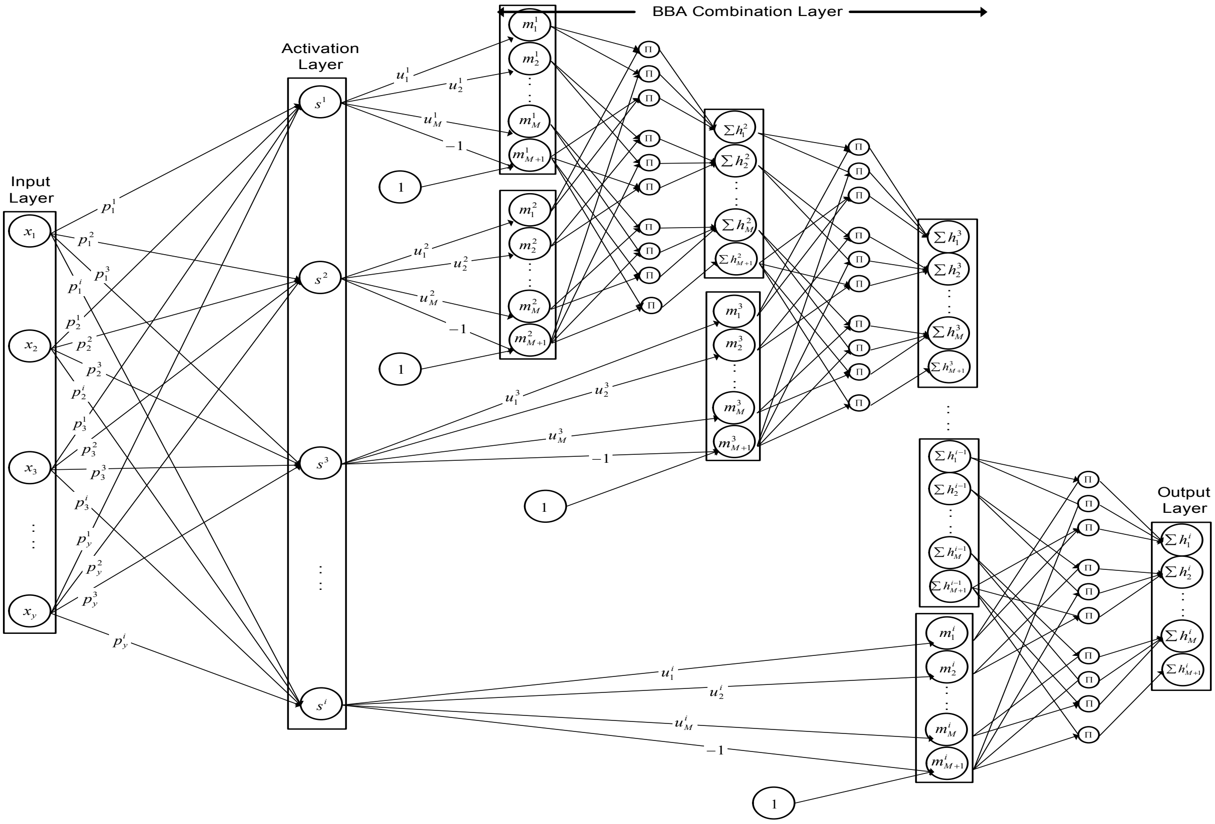

Figure 2. *Denoeux Belief Neural Network (DBNN) Architecture*

- ***x***: input to the network (test input or training input).
- ***y***: number of input nodes.
- ***p***: prototype, representing a number of ***k*** nearest previously trained input to the current tested input.
  - Prototypes are the representation of *the **k**-nearest neighbors of the trained input to the test input* in a simpler form to avoid computational complexity. These nearest neighbors are selected from the entire collection of the trained data. Where ***k*** is the number of nearest neighbors we want. However, in this network we need not set the value of ***k*** because it is not used since the neighbors are already represented by the prototypes, instead we set the number of prototypes we want. During training, ***p*** is adjusted to minimize the output error.
- ***i***: number of prototypes for each input node (we set this value).
  - There are a total of ***i*** x ***y*** number of prototypes in the network. The value of ***i*** must be less than the number of available training samples. The higher the value of ***i***, the easier it is for the network to converge, but the accuracy of test input classification may be affected.
- ***j***: current prototype number under consideration, as in $\boldsymbol{p^j}$.
- ***s***: the activation function.
  - For $1 \leq \boldsymbol{j} \leq \boldsymbol{i}$: $s^j = \alpha^j \exp(-(\eta^j)^2 (d^j)^2)$

    where,

    η: a parameter which is adjusted to minimize the output error during training.

    d: the distance between the tested input, ***x*** to the prototype, ***p***

$$\vec{d}^{\,j} = \left\| \vec{x} - \vec{p}^{\,j} \right\|$$

α: a parameter which is adjusted using ***ξ*** during training to minimize the output error.

$$\alpha^{j} = \frac{1}{1 + \exp(-\xi^{j})}$$

- ***M***: number of output classes.
- ***q***: current output class under consideration.
- u_q^j: a weight which represents the degree of membership for prototype ***j*** to output class ***q***.
  - If $u_q^j = 1$, then prototype ***j*** has full membership to class ***q***. There are ***i*** x ***M*** number of ***u***-weights in the network.
  - For 1 ≤ ***q*** ≤ ***M***:

    for 1 ≤ ***j*** ≤ ***i***:

$$u_q^j = \frac{(\beta_q^j)^2}{\sum_{k=1}^{M} (\beta_k^j)^2}$$

where,

β: a parameter which is adjusted during training to minimize the output error.

- m_q^j: the BBA mass.
  - It is the product of weight, u_q^j and activation function, ***s^j***.
  - For 1 ≤ ***j*** ≤ ***i***:

    for 1 ≤ ***q*** ≤ ***M***:

$$m_q^j = u_q^j s^j$$

for ***q*** = ***M*** + 1:

$$m_q^j = 1 - s^j$$

- h_q^j: the conjunctive combination of the BBAs.
  - For ***j*** = ***i***:

    for 1 ≤ ***q*** ≤ (***M*** + 1):

$$h_q^j = m_q^j$$

  - For 2 ≤ ***j*** ≤ ***i***:

    for 1 ≤ ***q*** ≤ ***M***:

$$\begin{aligned} h_q^j = {} & h_q^{j-1} m_q^j + h_q^{j-1} m_{M+1}^j + h_{M+1}^{j-1} m_q^j \\ & + h_{M+1}^{j-1} m_q^j \end{aligned}$$

for ***q*** = ***M*** + 1:

$$h_q^j = h_q^{j-1} m_q^j$$

- The normalized output of the network, $\boldsymbol{o_q}$ is denoted by,
  - for 1 ≤ ***q*** ≤ ***M***:

$$o_q = \frac{h_q^i}{\sum_{k=1}^{M+1} h_k^i}$$

  - The class ***q*** with the highest $\boldsymbol{o_q}$ value is selected as the prediction output.

- Output error for the training sample, ***z*** is given by,

  $$E_z = \frac{1}{2}\sum_{q=1}^{M}\left[\left(o_q + \frac{o_{M+1}}{M}\right) - T_q\right]^2$$

  where,

  T$\boldsymbol{T_q}$: the target output for class ***q***.

  $$o_{M+1} = \frac{h^i_{M+1}}{\sum_{k=1}^{M+1} h^i_k}$$

- Mean output error for ***N*** number of training samples is given by,

  $$E = \frac{1}{N}\sum_{z=1}^{N} E_z$$

  - ***E*** is minimized with respect to ***p***, ***η***, ***ξ*** and ***β***. The derivatives of ***E*** with respect to the parameters are provided in reference 7. Convergence to a local minimum of the error function can be ensured using iterative gradient-based optimization procedures described in reference 8.

3. Experiment and Results

In addition, a set of 73 proteins collected from another five classes was also used in one experiment (Table 2). Using a procedure similar to the one described above for the data set with 9 protein families, each of the 73 protein sequences was represented using 9 binary attributes with each attribute denoting the presence or absence of a motif. Performance compared to existing statistical and neural network techniques. It has proved extremely robust to strong changes in the distribution of input data. This advantage is extremely useful in the protein function prediction problem as the input data is volatile. Furthermore, it can reject the pattern under consideration if the associated uncertainty is too high, thus allowing implementing efficient novelty detection procedures.

Table 2: *The training sets of the 5 protein classes considered in this study.*

Family Document	Class
PDOC00360	Poly [ADP-Ribose] Polymerase, PPZF
PDOC00295	DNA Ligase, LIGASE
PDOC00605	Guanine Releasing Factor, GRF
PDCO50003	Cytoskeletal protein, CYTO
PDOC00463	Yeast Transcription activator, ACT

Five hundred and eighty five proteins belonging to one of the 10 classes (the false positive proteins from all of the 10 protein classes) were used in this experiment. Subsets of proteins were randomly picked from the 585-protein pool as the training samples. The sizes of the training sample sets were 11, 20,29,58,117,175,234,294,351, and 585 proteins. For a given training set size, the experiment was repeated three times using a different randomly sampled training set in each case. After DBNN classifier was built using a training set, all 585 proteins were used as the test set to determine classification accuracy of the resulting DBNN. The results shown in Figure 3 indicate that with only 10% of the total protein samples, DBNN could be constructed to classify proteins with an accuracy of 95%.

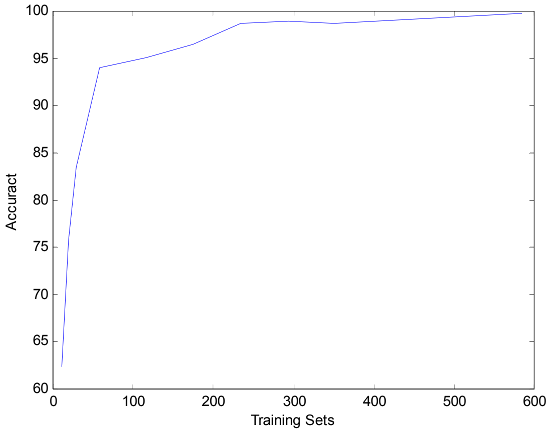

Figure 3: *Effect of training set size on classification accuracy.*

Each protein class, defined according to a Prosite documentation entry, it represented by one or more characteristic motifs. On the other hand, each motif is associated with a unique documentation entry, i.e., a protein class. Analysis of the DBNN used in this experiment indicated that the characteristic motifs of a protein class played a critical role in classification. On the surface, this might raise the question as to whether DBNN offer anything beyond a simple query of the Prosite database with the characteristic motif. However, a closer examination of the DBNN used by the algorithm indicates that there are situations in which the combinations of motifs that are used by the DBNN for separating the various families are different from the documented characteristic motifs for the corresponding families. Furthermore, the false positives generated by the DBNN are significantly fewer than those resulting from a Prosite search using the characteristic motif for each family. There were totally 11 false positive proteins from the 9 classes based on querying the Prosite database. The number of false positives resulting from the use of the DBNN trained using training sets of different sizes is shown in Figure 4.

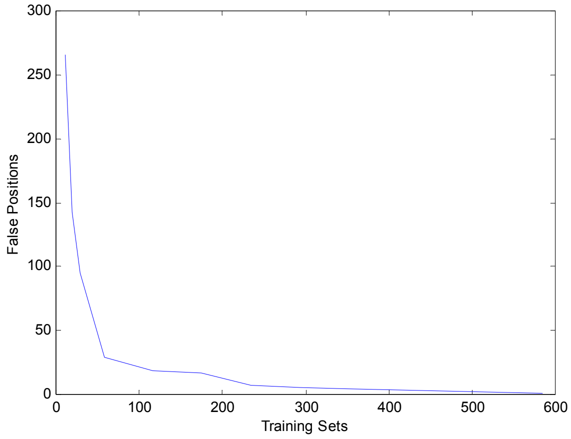

Figure 4: *The number of false positives resulting from the use of the DBNN trained using training sets of different sizes.*

The results show that the number of false positive classifications using DBNN falls below that resulting from Prosite search using characteristic motifs for training set sizes greater than or equal 40% of the data set. The number of false positives approaches zero as the fraction of the data set used for training approaches 100%. This suggests that DBNN program in fact discovers regularities among protein sequences that belong to a functional family that are not captured explicitly by their characteristic motifs as documented in the Prosite database.

To further explore this issue, a second data set of 73 protein sequences drawn from five classes (see Materials and Methods section for details) were used to build a DBNN classifier. The protein classes were chosen such that there were significant overlaps among the families in terms of their motif composition. For example, motif PS50010 (GRF_DBL) is present in proteins belonging to both classes PDOC00605 (GRF) and PDOC00360 (PPZF). In this scenario, querying the Prosite database with a single characteristic motif would result in a high rate of false positives. However, the DBNN classifier built by using randomly sampled training instances from this data set resulted in highly accurate assignment of sequences to the data set, the classification exceeded 96% when the size of the training set was greater than or equal to 22 (Figure 5). When the classifier was trained with 58 or more sequences (representing 80% or more of the data set) every sequence in the data set was correctly assigned to the corresponding functional family by the resulting sample DBNN constructed using a training set of size 58 is shown in Figure 3. DBNN distinguishes proteins belonging to class PDOC00360 from those belonging to class PDOC00605 based on the presence of PS50064 (PARP_ZN_FINGER_2) motif in the former but not in the latter although both families contain the PS50010 (GRF) motif.

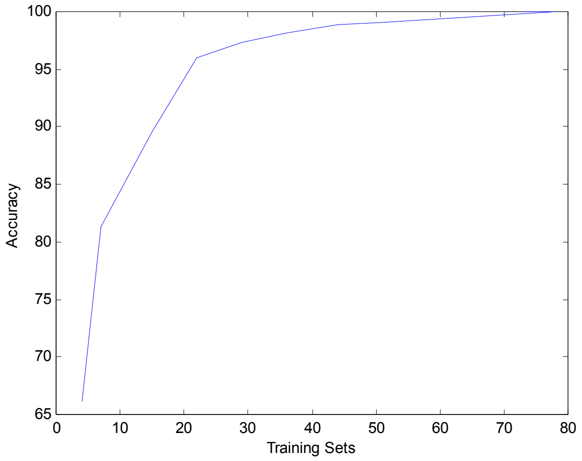

Figure 5. *Result of classifying proteins containing common motifs.*

The previous experiments demonstrate the effectiveness of the proposed approach in constructing fairly accurate models that capture regularities that help to accurately classify sequences belonging to different functional families. The extracted regularities are in form of combinations of motifs that are present or absent in the respective sequences. The accuracy of the resulting classifier exceeds that obtained by querying the Prosite database with the characteristic motif for each family. However, the real utility of the data-driven approach to building classifiers for functional classification of protein sequences would be in assigning novel sequences (with unknown function) to one of the known functional families. Conclusive demonstration of this would entail verifying the predictions of the classifier through biological experiments. However, we can assess the usefulness of the proposed approach in this context by systematic computational experiments where the predictions given by DBNN are compared with the (known) correct classifications on a part of the data set that is not used in training.

4. Discussion and Future Directions

Translating the recent advances in high throughput data acquisition technologies in biological sciences into fundamental gains in scientific understanding of biological processes calls for the development of sophisticated computational tools for characterization and prediction of macromolecular structure-function relationships. In this paper, we have presented an application of the DBNN learning algorithm for building protein sequence classifiers for assigning protein sequences to one of several functional families using a training set of sequences that are labeled with their corresponding functional families. The experimental results presented in this paper show that resulting DBNN classifiers are able to *generalize* well on test *sequences* that were not part of the training set. Furthermore, DBNN provides more accurate models of protein functional families than those based on *characteristic motifs* for some of the families documented in the Prosite database. Examination of the resulting DBNN indicates that the algorithm is able to discover from the data, the *presence or absence of combinations of subsets of motifs* that distinguish sequences belonging to each functional family from sequences belonging to other functional families represented in the training data. In particular, DBNN is able to identify *interactions* among motifs that can be quite far apart from each other with respect to their positions in the sequence. Such interactions might have a critical influence on the 3-dimensional structure and function of the protein.

Like any data driven technique, the proposed approach relies on the availability of representative sequences corresponding to proteins with known function for building the classifier. When such data is available, the proposed approach can be quite effective in assigning putative functions to novel sequences. This can serve as a useful source of information for guiding focused biological experiments.

Future work involves incorporating biological information into the model. Another direction for the future work involves systematic comparison of different machine learning algorithms for building predictors of protein function from sequence data; evaluation of the effectiveness of alternative approaches to motif detection in conjunction with different learning algorithms for building such predictors; and integration of the resulting tools with visualization routines for exploratory analysis of macro-molecular structure-function relationships.

References

1. Denoeux, T. (2000) “A neural network classifier based on Dempster-Shafer theory”. *IEEE transactions on Systems, Man and Cybernetics A,* 30(2):131-150.
2. Logan, B., P. Moreno, B. Suzek, Z. Weng and S. Kasif (2001). “A Study of Remote Homology Detection”. *Technical Report.*
3. Dayhoff, M. O.; Barker, W. C.; Hunt, L. T. (1983). "Establishing Homologies in Protein Sequences," *Methods in Enzymology*, **91,** 524.
4. Hofmann K., Bucher P., Falquet L., Bairoch A. (1999) *The PROSITE database, its status in 1999* Nucleic Acids Res. 27:215-219.
5. Bateman, A. Birney, E.., Durbin, R., Eddy, S., Howe, K., and Sonnhammer, E. (2000) *Nucleic Acids Research*, 28:263-266.
6. Attwood, T.K. and Beck, M.E. (1994) PRINTS - A protein motif finger- print database. Protein Engineering, 7 (7), 841-848.
7. Denoeux, T. (1995) “A k-nearest neighbor classification rule based on Dempster-Shafer theory”. *IEEE Transactions on Systems*, Mon and Cybernetics, 25(50):804-813.
8. Hudak, J. and McClure,M.A.(1999). “A Comparative Analysis of Computational Motif Detection Methods”. *Pacific Symposium on Biocomputing* 4:138-149.

Journal Home Page