ISSN 0974 - 9373

Vol. 15(2011) Special Issue 2 **Journal of International Academy of Physical Sciences** pp. 515-524

# Recursive Neural Networks Based Recognition of Protein Folds on Account of Residue Correlation within Protein Sequences*

**Pooja Mishra**
Center of Bio-Informatics, University of Allahabad, Allahabad
E-mail: pooja.mishra0806@gmail.com

**P. N. Pandey**
Department of Mathematics, University of Allahabad, Allahabad

(Received June 14, 2011)

**Abstract:** "Protein sequences can be regarded as slightly edited random strings"[1]. We applied the methods of estimating residue correlation within the protein sequences. First we use the mutual information (MI) of adjacent residues, and improve the long range correlation between nonadjacent residues by defining the mutual information vector (MIV) of each protein sequence. The correlation is based on residue hydropathy rather than protein-specific interaction. Like this we calculated the MIV of each protein sequence, and these MIV are further give to recursive neural network to obtain the classification of protein sequence. The modeling power of MIV was shown to be significantly better, reaching the level where proteins can be classified without alignment information.

**Keyword:** protein folds classification, neural networks, mutual information in sequences.

## 1. Introduction

Protein fold recognition is the basis in protein structure discovery process, especially when traditional sequence comparison methods fail to yield convincing structural homologies. Although many methods have been developed for protein fold recognition, their accuracies remain low. Several pre-defined methods for protein fold recognition assumes that the number of protein folds in the universe is limited and therefore the protein folds recognition can be viewed as the fold classification problem, where a query protein can be classified into one of the known folds. In this classification scheme one needs to identify fold-specific features, which can discriminate between different folds.

*Paper presented in CONIAPS XIII at UPES, Dehradun during June 14-16, 2011.

A protein can be viewed as a string composed from the 20 symbol of amino acid alphabet or, alternatively, as the sum of their structural properties, for example, residues involved in forming alpha helix, beta sheets or participating in other secondary structure and residues involved in solvent accessibility such as buried/exposed residues. Protein sequences contain sufficient information to construct secondary and tertiary protein structures. Most methods for predicting protein structure rely on primary sequence information by matching sequences representing unknown structures to those with known structures. Thus, researchers have investigated the correlation of amino acids within and across protein sequences [2-3]. Despite all this, in terms of character strings, proteins can be regarded as slightly edited random strings[1]. Previous study has shown that residue correlation can provide biological insight, but that MI calculations for protein sequences require careful adjustment for sampling errors. An information-theoretic analysis of amino acid contact potential pairings with a treatment of sampling biases has shown that the amount of amino acid pairing information is small, but statistically significant [2]. Another recent study by Martin et al. [3] showed that normalized mutual information can be used to search for coevolving residues. MIV has significantly better modeling power of proteins than MI, demonstrated in the protein sequence classification experiment [4]. In the present work, we used the protein family information from Pfam [5]. To model sequences, each protein sequence is associated to mutual information vector (MIV) where each entry of MIV represents the MI estimation for amino acid pairs separated by a particular distance in the primary structure. We studied two different properties of sequences: structural properties and solvent accessibility. The RNN [6-7], which is characterized by higher computing ability than the BP network, is applied to classify the protein data. Experimental results show that RNN significantly improves the accuracy and reliability of classification.

## 2. Material and Methods

### 2.2 Dataset

The dataset used in this study is the Ding and Dubchak dataset (D-B dataset), which is same as that used in earlier studies[8,9]. The D-B dataset contains 311 and 383 proteins for training and testing, respectively (http://cdr.lbl.gov~cding/protein). This dataset has been termed such that, in the training set, no two proteins have more 35% sequence identity to each other and each fold have seven, or more proteins; and in the test set, proteins have <40% identity to the proteins of the training set. According to SCOP classification[10], the proteins used for training and testing belong to 27

different folds representing all major structural classes: all α, all β, α/β, α+β and small proteins.

### 2.2.1 Mutual information (MI) content

MI content is used to estimate correlation in protein sequences to gain insight into the prediction of secondary and tertiary structures. MI is a measure of correlation from information theory [11] based on entropy, which is a function of the probability distribution of residues. We can estimate entropy by counting residue frequencies. Entropy is maximal when all residues appear with the same frequency. MI is calculated by systematically extracting pairs of residues from a sequence and calculating the distribution of pair frequencies weighted by the frequencies of the residues composing the pairs. By defining a pair as adjacent residues in the protein sequence, MI estimates the correlation between the identities of adjacent residues.

### 2.2.2 Mutual information

The entropy of a random variable X, H(X), represents the uncertainty of the value of X. *H(X)* is 0 when the identity of X is known, and *H(X)* is maximal when all possible values of X are equally likely. The mutual information of two variables *MI(X, Y)* represents the reduction in uncertainty of X given Y, and conversely, *MI(Y, X)* represents the reduction in uncertainty of Y given X:

$$MI(X,Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X) \ . \tag{4}$$

When X and Y are independent, *H(X | Y)* simplifies to *H(X)*, so *MI(X, Y)* is 0. The upper bound of *MI(X, Y)* is the lesser of *H(X)* and *H(Y)*, representing complete correlation between X and Y:

$$H\left(X \mid Y\right) = H\left(Y \mid X\right) = 0 \ . \tag{5}$$

We can measure the entropy of a protein sequence S as

$$H(S) = -\sum_{i \in \sum A} P(x_i) \log_2 P(x_i), \tag{6}$$

where ΣA is the alphabet of amino acid residues and $P(x_i)$ is the marginal probability of residue i. In Section 3.3, we discuss several methods for estimating this probability. From the entropy equations above, we derive the MI equation for a protein sequence $X = (x^1, \ldots\ldots, x^N)$:

$$(7) \qquad MI = \sum_{i \in \sum A} \sum_{j \in \sum A} P(x_i, x_j) \log_2 \left( \frac{P(x_i, x_j)}{P(x_i) P(x_j)} \right),$$

where the pair probability $P(x_i, x_j)$ is the frequency of two residues being adjacent in the sequence.

### 2.2.3 Normalization by Joint Entropy

Since *MI(X,Y)* represents a reduction in *H(X)* or *H(Y)*, the value of *MI(X,Y)* can be altered significantly by the entropy in X and Y. The MI score we calculate for a sequence is also affected by the entropy in that sequence. Martin et al. [3] propose a method of normalizing the MI score of a sequence using the joint entropy of a sequence. The joint entropy, or *H(X, Y)*, can be defined as

$$(8) \qquad H(X,Y) = \sum_{i \in \sum A} \sum_{j \in \sum A} P(x_i, x_j) \log_2 P(x_i, x_j),$$

and is related to *MI(X,Y)* by the equation

$$(9) \qquad MI(X,Y) = H(X) + H(Y) - H(X,Y).$$

The complete equation for our normalized MI measurement is

$$(10) \qquad \frac{MI(X,Y)}{H(X,Y)} = - \frac{\sum_{i \in \sum A} \sum_{j \in \sum A} P(x_i, x_j) \log_2 \left( P(x_i, x_j) / P(x_i) P(x_j) \right)}{\sum_{i \in \sum A} \sum_{j \in \sum A} P(x_i, x_j) \log_2 P(x_i, x_j)}.$$

### 2.2.5 Distance Mutual Information Vector (MIV)

Protein exists as a folded structure, allowing nonadjacent residues to interact. Furthermore, these interactions help to determine that structure. For this reason, we use MIV to characterize nonadjacent interactions. Our calculation of MI for adjacent pairs of residues is a specific case of a more general relationship, separation by exactly *d* residues in the sequence.

**Definition 1:** For a sequence S = ($s^1$……..$s^N$), mutual information of distance d, MI(d) is defined as

$$(11) \qquad MI(d) = \sum_{i \in \sum A} \sum_{j \in \sum A} P_d(x_i, x_j) \log_2 \left( \frac{P_d(x_i, x_j)}{P(x_i) P(x_j)} \right).$$

The pair probabilities, $P_d(x_i, x_j)$, are calculated using all combinations of positions $s^m$ and $s^n$ in sequence S such that

$$m + (d + 1) = n, \qquad n \leq N. \tag{12}$$

A sequence of length N will contain N − (d + 1) pairs.

**Definition 2:** The mutual information vector of length k for a sequence X, $MIV_k(X)$, is defined as a vector of k entries, {MI(0),…….., MI(k − 1)}.

### 2.2.6 Sequence alphabets

The alphabet chosen to represent the protein sequence has two effects on our calculations. First, by defining the alphabet, we also define the type of residue interactions we are measuring. By using the full amino acid alphabet, we are only able to find correlations based on residue-specific interactions. If we instead use an alphabet based on hydropathy, we make correlations based on hydrophilic/hydrophobic interactions. Second, altering the size of our alphabet has a significant effect on our MI calculations. In our study, we used three different alphabets: a set of 20 amino acids residues, $\boldsymbol{\Sigma_A}$, a secondary structure-based alphabet, $\boldsymbol{\Sigma_S}$, and a solvent accessibility-based $\boldsymbol{\Sigma_X}$, derived from grammar complexity and syntactic structure of protein sequences[12] (see table 1 for mapping $\boldsymbol{\Sigma_A}$ to $\boldsymbol{\Sigma_S}$ and $\boldsymbol{\Sigma_X}$), an example of MIV's calculated for single amino acid composition.

Table 1: Amino acid partition based on their secondary structure and solvent accessibility

| ***Secondary structure based partition*** | |
|---|---|
| Secondary structure | Amino acids |
| Alpha helix | R, E,Q,H,K |
| Beta strand | C,I,M,F,W,Y,V,L |
| Random coil | N,D,S,T,P,A,G |
| ***Solvent accessibility based partition*** | |
| Buried | C,I,M,F,W,Y,V,L |
| Exposed | R,N,D,E,Q,H,K,S,T,P,A,G |

### 2.3.1 Recursive Neural Network

The RNN is taken as a basic classifier[6]. This network type consists of an input layer, a hidden layer, and an output layer. In this way, it resembles a three layer feed-forward neural network. However, it also has a context layer, in which the neurons hold a copy of the output of the hidden neurons. The value of each context neuron is used one time step later as an extra input

signal for all the neurons in the hidden layer. The addition of interior feedback network increases the capability of processing dynamic information of the network itself, and therefore makes the system have the ability to adapt to time-varying characteristics. Suppose there are r inputs, m outputs and n neurons, respectively, in the hidden layer and in the context layer. *u(k-1)* represents the inputs of Elman network; *x(k)* represents the outputs of the hidden layer; $x_c(k)$ represents the outputs of the context layer, and *y(k)* represents the outputs of Elman network. Then, its nonlinear state-space expression is

$$(13) \qquad \begin{cases} y(k) = g(w_2 x(k)), \\ x(k) = f(w_3 x_c(k) + w_1(u(k-1))),) = x(k-1), \\ x_c(k \end{cases}$$

where $w_1$ is the weight from input layer to hidden layer, $w_2$ from hidden layer to output layer and $w_3$ from context layer to hidden layer. *g* represents the transfer function of the output layer, which is usually a linear function. *f* represents of the hidden layer, *S* type function is commonly used and can be defined as

$$(14) \qquad f(x) = (1 + e^{-x})^{-1}.$$

Back propagation algorithm with momentum of variable learning rate is used here to modify the weight values and the error of the network is

$$(15) \qquad E = \sum_{i=1}^{m} (t_i - y_i)^2,$$

in which $t_i$ *(i*=1, 2, … *m)* are the output vectors of the object.

### 2.3.2 Parameters Setting

We have taken *d* from 0 to 19, this gives us a 20 dimensional vector. Thus, the input neurons correspond to 20. Eight neurons are used for the hidden layer. During the training process, the generalization error is estimated in each epoch on a validation set. If the error does not change in six consecutive epochs, the training of the network is terminated in order to avoid overfitting.

We use seven-fold cross-validation for training and evaluating the prediction performance, in which a data set is divided into seven subsets of approximately equal size. This means that the data is partitioned into training and test data in seven different ways. After training a classifier with a

collection of six subsets, the performance of the classifier is tested against the seventh subset. This process is repeated seven times so that every subset is once used as the test data. In the tests, it is run by seven times with intent to ensure the rationality of results, because Back Propagation algorithm over multilayer networks is only guaranteed to converge toward some local minimum and not necessarily to the global minimum error[13].

## 3. Result and Discussion

Several experiments were conducted to evaluate the proposed method. The classification accuracy was measured by counting the sensitivity and specificity rates. In all $K$-class classification problems, each protein family $S_k$ ($k = 1, \ldots, K$) was randomly partitioned into training and test sequences, with the training set being only a small percentage (5 - 10%) of the family dataset. The proposed method was evaluated using Five-fold cross validation on the D-B dataset of non-redundant protein chains.

Performance is assessed using a variety of standard measures including correlation coefficients area under the ROC curves. Accuracy at 5% FPR (false Positive Rate), Precision [TP/(TP+FP)] and Recall [TP/(TP+FN)]. The accuracy at 5% FPR is defined as [(TP+TN)/ (TP+FP+TN+FN)] when the decision threshold is set so that 5% of negative cases are above the decision threshold. Here, TP, FP, TN and FN refer to the number of true positives, false positives, true negatives and false negatives respectively.

To evaluate classification performance ROC (Recursive Operating Characteristic) analysis was used. More specifically, we used the $ROC_{50}$ curve which is a plot of sensitivity as a function of false positive for various decision threshold values until 50 false positives are found. The area under the ROC curve in this method computed on all regions is 0.878, shown in figure 1. An area of 1.00 would correspond to a perfect predictor and an area of 0.50 would correspond to random predictor. The results of RNN over D-B training and testing datasets is described in table 2. We have also compared our results to those of other predictors. Table table 3 shows our results in comparison to other predictors.

Table 2: Results of RNN over D-B dataset

| ***Dataset*** | ***Correlation coefficient*** | ***ROC area*** | ***Accuracy*** | ***Precision*** | ***Recall*** |
|---|---|---|---|---|---|
| Training dataset | 0.589 | 0.878 | 92.8% | 75.4% | 38.8% |
| Test dataset | 0.255 | 0.789 | 94.5% | 22.1% | 25.9% |

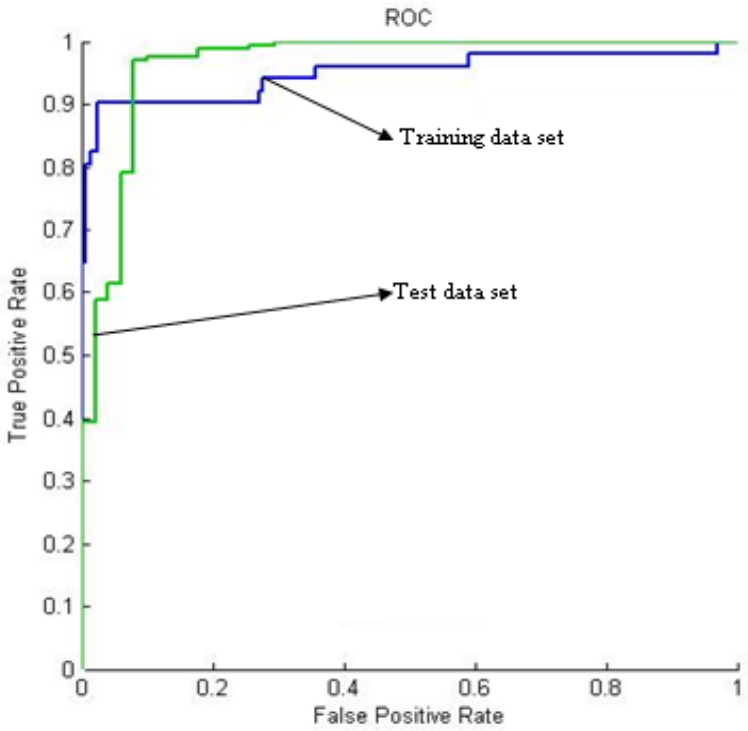


Figure 1: ROC curve of RNN on the set of 723 protein chains

Table 3: Comparison of RNN with all other machine learning algorithms

| ***Machine learning algorithms*** | ***Accuracy (%)*** | | |
|---|---|---|---|
| | $\Sigma_A$ | $\Sigma_S$ | $\Sigma_X$ |
| Least Hamming distance [14] | 85.22 | 79.23 | 87.2 |
| Least Euclidean distance [15] | 84.56 | 82 | 94 |
| ProtLock [16] | 79.43 | 83.6 | 91 |
| Covariant-discriminant [17] | 79.82 | **88.74** | **95.82** |
| Augmented covariant discriminant [18] | **88.53** | 87 | 88.6 |
| Support vector machines | 66.1 | 80.7 | 86.85 |
| SLLEc & KNN | 72.67 | 88.43 | 92.67 |
| **Recursive Neural Networks** | **89.3** | **90.8** | **93.3** |
| Fuzzy KNN | 73.69 | 85.61 | 88.92 |
| Support vector machines | 81.98 | 78.12 | 86.27 |

## 5. Conclusion

The advantages in using the representation of MIVs to classify the protein folds are: (1) allowing us to use a discrete model to deal with a problem involving many sequences with extreme variation in length; (2) able to incorporate a considerable amount of sequence order effects that are hidden in long and complicated protein sequences; and (3) providing a flexible mathematical frame to invite various novel approaches. The current Elman RNN approach is just one of them. Nevertheless, as demonstrated by the overall success rates, it is a quite promising one. Furthermore, it is intriguing to note that the Elman RNN approach as introduced here may also have a positive impact in improving the prediction quality for protein classifications.

## References


1. O. Weiss, M. A. Jimenez-Montano, and H. Herzel, "Information content of protein sequences," *Journal of Theoretical Biology*, **206(3)** (2000) 379–386.
2. M. S. Cline, K. Karplus, R. H. Lathrop, T. F. Smith, R. G. Rogers Jr., and D. Haussler, "Information-theoretic dissection of pairwise contact potentials," *Proteins: Structure, Function and Genetics*, **49(1)** (2002)7–14.
3. L. C. Martin, G. B. Gloor, S. D. Dunn, and L. M. Wahl, "Using information theory to search for co-evolving residues in proteins," *Bioinformatics* **21(22)** (2005) 4116–4124.
4. Chris Hemmerich and Sun Kim, A Study of Residue Correlation within Protein Sequences and Its Application to Sequence Classification, *EURASIP Journal on Bioinformatics and Systems Biology*, (2007) doi:10.1155/2007/87356.
5. Alex Bateman, Lachlan Coin, Richard Durbin, Robert D. Finn, Volker Hollich1, Sam Griffiths-Jones, Ajay Khanna, Mhairi Marshall, Simon Moxon, Erik L. L. Sonnhammer, David J. Studholme, Corin Yeats and Sean R. Eddy "The Pfam protein families database," *Nucleic Acids Research* **32** (2004) D138–D141.
6. J. Elman, Finding structure in time. *Cog. Sci*. **14**(1990)179-211.
7. Shi, X. H., Liang, Y. C., Lee, H. P., Lin, W. Z., Xu, X. and S. P. Lim, Improved elman networks and applicationsfor controlling ultrasonic motors. *Appl. Artif. Intell*., **18** (2004) 603-629.
8. H. B. Shen and K. C. Chou, Ensemble classifier for protein fold pattern recognition, *Bioinformatics*. **22(14)**(2006) 1717-1722.
9. Kuo-Chen Chou, Prediction of protein cellular attributes using pseudo amino acid composition, *PROTEINS: Structure, Function, and Genetics*, **43**(2001) 246-255.
10. A. G. Murzin, S.E. Brenner, T. Hubbard, C. Chothia, SCOP: a structural classification of proteins database for the investigation of sequences and structures, *J Mol Biol.* 7; **247(4)**(1995)536-40.
11. T. M. Cover and J. A. Thomas (1991) Elements of Information Theory, *Wiley-Interscience,* New York, NY, USA.

12. M. A. Jimenez-Montano "On the syntactic structure of protein sequences and the concept of grammar complexity," *Bulletin of Mathematical Biology,* **46(4)** (1984)641–659.

13. C. Bishop, *Pattern recognition and machine learning. Springer*, New York, USA, (2006)225-284.

14. P. Y. Chou, G.D. Fasman, *Prediction of protein structural class from animo acid composition. In: Prediction of Protein Structure and the Principles of Protein Conformation* (Fasman, G.D., ed.), Plenum Press, New York, (1989)549–586

15. H. Nakashima, K. Nishikawa, Discrimination of intracellular and extracellular proteins using amino acid composition and residue-pair frequencies, *J. Mol. Biol.* **238** (1994) 54-61.

16. J. Cedano, P. Aloy, J.A. Pérez-Pons, E. Querol, Relation between amino acid composition and cellular location of proteins, *J. Mol. Biol.* **266**(1997) 594-600.

17. K. C. Chou and D. W. Elrod. Prediction of membrane protein types and subcellular locations, *Proteins.* **34**(1999)137-153.

18. P. Baldi and S. Burnak 1998) *Bioinformatics: the machine learning approach*, MIT Press, Cambridge, MA.