Classification of Protein 3D Structures Using Artificial Neural Network

Hui Li, Chunmei Liu, Legand Burge, and William Southerland

***Abstract*—Three dimensional (3D) protein structures determine the function of a protein within a cell. Classification of 3D structure of proteins is therefore crucial to inferring protein functional information as well as the evolution of interactions between proteins. In this paper, we utilized the artificial neural network (ANN) paradigm to classify the protein structures. The approach equally divides a 3D protein structure into several parts and then extracts statistical features from each part. The different parts in spatial domain are thus viewed as the feature vectors which are input into the neural network. The computational experiments achieve better results in most cases than other currently existing approaches.**

***Index Terms*—Protein structure; classification; ANN.**

I. Introduction

Three dimensional (3D) structures of a protein are determined by the amino acid sequence. Protein functions depend on their structures. Generally, the determination of protein structures aims at inferring functional information of the protein. With the incredible increase of sequenced protein information and identified protein structure information, especially the extensively developed protein structure databases, the development of a systematical, automatic, efficient and comprehensive method to classify the protein structures is urgently needed.[1]

Various public databases such as PDB [1], DIP [2] and BioGrid [3] have provided a snapshot of protein structures and their interactions at a great level of details than other experimental methods. Protein structure classifications such as CATH [4] and SCOP [5] are most useful. CATH and SCOP are primary and secondary structure based classifications which rely on experts to manually check the classifications. Such classifications organize protein structures into families. Those primary structure based methods may include well-conserved information in evolution.

Most of the existing classification methods are based on shapes, i.e., calculating the distances between pairs of Cα atoms such as RMSG [6]. Consequently, they have to deal with tens of thousands of structures for each protein. These approaches consume significant computation space.

Machine learning methods to cluster and classify protein structures to understand protein structures have recently become a very active area of research. In this paper, we use a statistical method as feature vectors to classify the protein structures. Statistical methods include Shape Histograms proposed by Anskerst [7], Shape Distribution by Osada [8], Ohbuchi put forward Shape Function and eigen-CSS proposed by Mark. Since the statistics methods are based on overall feature extractions, using statistical methods has a high robust of the boundary noise as well as good performance for rough classification. We employ artificial neural networks (ANN) to systematically develop a comprehensive view of protein structures to infer protein functional information. Since ANN [9] is advantageous in multiple classifications, we use it to predict the protein 3D structures.

In the following section, we first introduce the preprocessing of our statistical model for protein structure classification. We then introduce the classification mechanism and the feature extraction. Then the experimental results demonstrate that the ANN is capable of classifying the 3D protein structures. Finally, the conclusion and future work are discussed.

II. Feature Extraction

A. Preprocessing of the Protein Structure Data

Since the number of molecules for each protein is different, we need to normalize them into one size for comparing them. The PDB provides a number of coordinates for *C-alpha* atoms which can outline the 3D shape for the whole protein. Given a 3D protein structure $M_P = \{a_1, a_2, \ldots a_n\}$ where a_i represents the *C-alpha* atom in the protein, we use transformation and rotation to normalize the protein structure. The process can be denoted as Equation 1.

$$S_p(M_p) = R\left(N\left(M_p\right) - T\right) \quad (1)$$

where T is the transformation matrix on the original 3D matrix computed by the maximum distance (d_{max}) between surface points and the minimum distance (d_{min}), and R is the rotation matrix of the adjustment after the T transformation of M_P. After preprocessing for the protein structure, we put the 3D protein structure into a unit sphere.

Based on the study of Ankerst et al [9], we divide the unit sphere into 30 equal parts and the feature selection is executed on each partition for the unit sphere, and then we count the number of atoms for each segment. Subsequently,

Manuscript received September 25, 2012; revised November 11, 2012. This work was supported by NSF CAREER (CCF-0845888) (Li and Liu), NSF Science & Technology Center grant CCF-0939370 (Li, Liu, and Burge), and 2 G12 RR003048 from the RCMI program, Division of Research Infrastructure, National Center for Research Resources, NIH (Southerland).

Hui Li, Chunmei Liu, and Legand Burge are with the Department of Systems and Computer Science, Howard University, Washington, DC 20059, USA (e-mail: hli@scs.howard.edu; chunmei@scs.howard.edu; blegand@scs.howard.edu.).

William Southerland is with the Department of Biochemistry and Molecular Biology Howard University, Washington, DC 20059, USA (e-mail: wsoutherland@fac.howard.edu).

the unit sphere can be represented as $S=\{S_1, S_2,\ldots S_i, S_n\}$, where S_i is the segment of the whole protein. P_{sm} is the sum of atoms in the whole protein S, V_i is the ratio of the number of atoms in each segment S_i over the total number of atoms in the protein. It is denoted as Equation 2 :

$$V_i = S_i / P_{sm} \tag{2}$$

where $i=1,2,\ldots,n$. We get a feature vector $V=\{V_1,V_2,\ldots,V_n\}$. The feature vector reflects the spatial characteristics of the protein structure shown as Fig. 1.

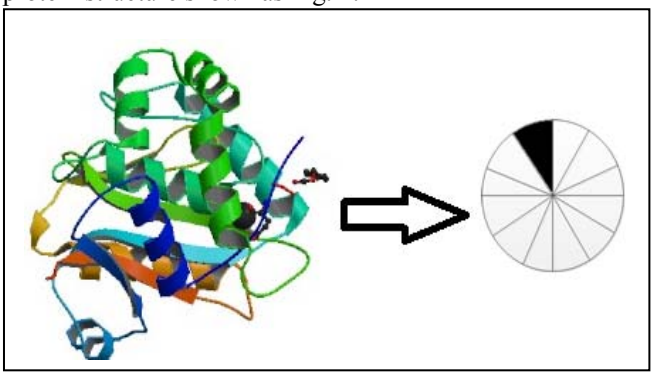

Fig. 1. The conversion from a 3D protein structure to a statistical model

B. Classification

1) *Mechanism of the artificial neural network mode.*

A BP (Back Propagation) network [9] is an error back propagation algorithm to train the multilayer network. It is one of the most widely used neural network model. Since BP is able to deal with multi-features, we employ BP to classify the protein structures. BP neural network can learn and store a lot of input-output mode mapping without prior revealing of the mathematical equations describing the mapping. The learning rule uses the steepest descent method and back-propagation to adjust the network weights and thresholds to obtain the minimum error of the network. BP neural network model topology includes input layer (input), hidden layer (hide layer) and output layer.

2) *ANN structure*

Our artificial neural network is made up of an input layer, one hidden layer, and an output layer. Each layer is composed of computing neuron unit. Given a set of 3D proteins, we use the above statistical based method on each protein structure, and then extract the statistic features $V=\{ v_1,v_2, v_3, \ldots v_n\}$ which are made up of the neuron units in the input layer. The architecture of the ANN for our method is shown in Fig. 2.

$$Input = \{v_1, v_2, v_3, \cdots v_n\} \tag{3}$$

Each neuron of the input layer is represented as the density associated with the segment in the protein structure and the value of neuron unit within hidden layer is calculated according to the corresponding connection weights. The weights associated to the input layer, hidden and output layer connections are initialized with small random value which is less than 0.1. Net is the value of the neurons in the hidden layer which is computed by Equation 4:

$$Net = v_1 w_1 + v_2 w_2 + \ldots v_n w_n \tag{4}$$

where $w_1, w_2,\ldots, w_n$ represent the weights between the hidden layer and output layer, and $v_1,v_2,\ldots,v_n$ are the values of the neurons in the input layer.

The network output is denoted as different categories of the 3D protein structure folds. Since we have totally 27 classes, five binary values can be used to represent the output folds. The value of the output layer is indicated as Equation 5.

$$E_o = \{o_1, o_2, o_3, o_4, o_5\} \tag{5}$$

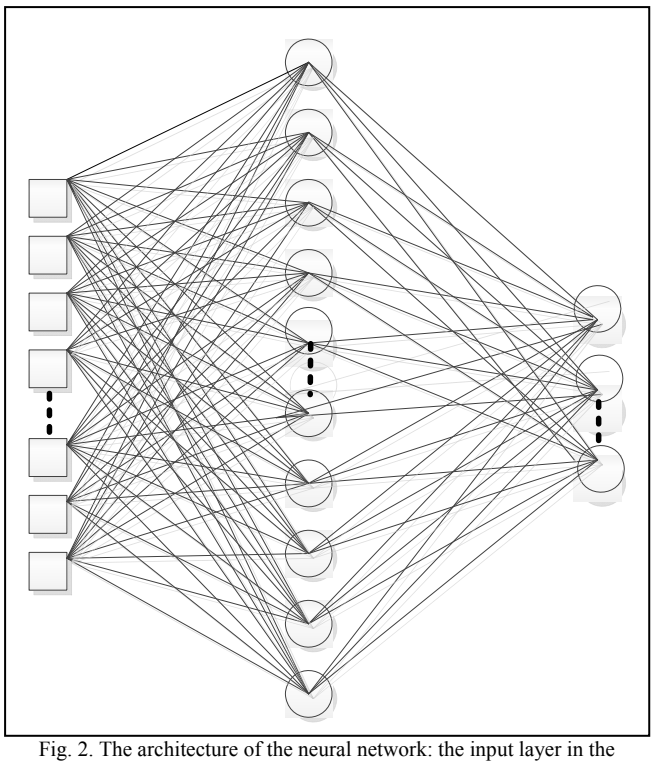

Fig. 2. The architecture of the neural network: the input layer in the network is represented by circles and the connections between units by arrows. The input layer is made up of 30. The hidden layer includes 70 neurons in total. The output layer is denoted as types of protein folds.

Subsequently, the neurons of the output layer produce a result from the hidden layer and the input layer according to a sigmoidal activation function shown as Equation 6

$$Q_i = 1/(1+e^{-t}) \tag{6}$$

During BP neural network training, the value of neurons within the output layer will update by adding the weights within the input layer. Each neuron will be activated by the sigmoidal function. The BP network adjusts its connection weights between input and output according to the difference between the expected output value and the experiment output value. The process of the adjustment repeats until the error meets the requirement, and then we can obtain the parameter of the ANN model.

The signal is the input sample data transferred from the input layer and the hidden layer to the output layer and the process of the adjustment transfers the results from the output layer to the input layer.

III. Data Sets

In this paper, we use data from the spatial structures of Structural Classification of Proteins (SCOP) databases

which are based on the similarities of their amino acid and three-dimensional structures [10]. This database includes fold classification family that are connected to sequence similarity.

The training set contains 1411 samples and a testing set include1185 samples. According to the SCOP classification, the whole dataset consists of 27 types of folds.

IV. Results

A Dell server with 4 dual processors was used as the main hardware platform in the system. We used the ANN toolbox in MATLAB and a Java class to implement the method. Java class can invoke the ANN through JAVA GUI which provides friendly interface to set parameters in the ANN. Also java class is responsible for dynamic parsing protein structures from online databases when users input the query protein name.

In the training process of the ANN, all the weights associated to each layer are initialized using small values ranging in [0-1] and small values are generated using the pseudo-random number generator.

The values of neurons within the output layer are transformed from the float values which in the range of (0.0-1.0) into binary values. The neuron value is set to be 1 if the output is greater than 0.5. Otherwise, it is set to be 0.

The total error is calculated by a sigmoidal activation function on the output neurons according to their corresponding weights. We use the evolution of the total error to measure the performance of the ANN model in our method.

As shown in Table I. We use the accuracy to measure the performance of the neural networks. The accuracy represents the ratio between the number of correctly predicted protein fold classes and the total number of samples.

TABLE I: The Comparison for the Performance between SCOP and Our Method

Method name	Accuracy
SCOP	64%
Our method	71%

V. Conclusion

In this paper, we describe a novel approach to classify protein structures. First, we normalize the 3D protein structures and put them into a unit sphere. Second, we use a statistical method to extract the feature vector which reflects the shape of the protein structure. Finally, we assemble ANNs for protein structure classification based on the statistical feature vector. The experimental results show our method achieves 71% accuracy on protein structure classification.

References

[1] G. Joosten R. P, K. Joosten, G. N. Murshudov, and A. Perrakis PDB_REDO, constructive validation, more than just looking for errors. Acta Crystallogr D Biol Crystallogr. 2012 Apr; 68(Pt 4):484-96.

[2] B. Lehne and Schlitt T. *Protein-protein interaction databases: keeping up with growing interactomes*. Hum Genomics. 2009 Apr; 3(3): 291-7.

[3] B. J. Breitkreutz, C. Stark, T. Reguly, L. Boucher, A. Breitkreutz, M. Livstone, R. Oughtred, D. H. Lackner, J. Bähler, V. Wood, K. Dolinski, M. Tyers. The BioGRID Interaction Database: 2008 update. *Nucleic Acids Res*. 2008 Jan; 36(Database issue):D637-40. Epub 2007 Nov 13.

[4] L. C. Alison, S. Ian, L. Tony, C. R. Oliver, G. Richard, T. Janet, and A. Christine. "The CATH classification revisited—architectures reviewed and new ways to characterize structural divergence in super families," *Nucleic Acids Research*, 2009, 37(Database):D310-D314.

[5] A. Andreeva, D. Howorth, J. M. Chandonia, S. E. Brenner, T. J. P. Hubbard, C. Chothia, and A. G. Murzin. "Data growth and its impact on the SCOP database: new developments," *Nucleic Acids Research*, 2008, 36 (Database):D419~D425.

[6] H. Berman, K. Henrick, and H. Nakamura, "Announcing the worldwide Protein Data Bank," *Nature Structural Biology*, 2003,10(12):980~980

[7] L. Holm, S. Kaariainen, P. Rosenstrom, and A. Schenkel. Searching protein structure databases with DaliLite vol. 3. Bioinformatics, 2008,24(23):2780~2781

[8] V. N. Vapnik. "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, 1999, vol.10, no.5, pp.988-999.

[9] M. Ankerst, G. Kastenmüller, and H. P. Kriegel, et al. "3D Shape histograms for similarity search and classification in spatial databases[C] Lecture Notes in Computer Science," in *Proc. 6th International Symposium on Spatial Databases*, Hong Kong, China. Berlin: Springer Verlag, 1999, pp.5-8.

[10] K. Marsolo and S. Parthasarathy, Ding C. A multi-level approach to SCOP folds recognition. in *Proc. Fifth IEEE Symposium on Bioinformatics and Bioengineering*. Washington, DC: IEEE Computer Society, 2005. pp.57~64.

Hui Li is currently a Postdoctoral Fellow at the Department of Systems and Computer Science in Howard University. He received his PhD in Computer Science from Beijing School of Computer Science, University of Technology, Beijing in 2009. His research interests include computational biology, bioinformatics, pattern recognition and algorithm.

Chunmei Liu is currently an Associate Professor of the Department of Systems and Comp uter Science at Howard University. She got her Ph.D.. in Computer Science from the University of Georgia; Athens, GA in 2006. Her research interests include Graph Theory, Computational Biology, Algorithms, and Complexity.

Legand Burge is Professor and Chair of the Department of Systems and Computer Science at Howard University. Dr. Burge got his Ph.D. in Computer Science from Oklahoma State University in 1998.

He is interested in distributed computing, bioinformatics and machine learning.

Dr. Southerland is currently Professor of the Department of Biochemistry at Howard University, College of Medicine.

He is interest in molecular structure and biomedical imaging that include molecular computations, bioinformatics, structural NMR, and imaging analyses.