A Machine Learning Approach To Protein Fold Recognition

Ashwin Fernandes, Kavya Kotian, Chiraag Limaye and Karan Manghi
Supervisor: Mr. Uday Nayak

Information Technology
Don Bosco Institute of technology
Vidyavihar , Mumbai 400 077
Email: ashphoenix16@gmail.com, kbkotian@gmail.com,
chiraag.limaye7@gmail.com, karan$_m anghi@yahoo.in$

***Abstract*—Proteins are key functional units in living organisms and are involved in many biological processes in the cell. Protein folding is the physical process by which a protein chain acquires its native 3-dimensional structure. The phenomenon of protein folding is very important in Biology as a proteins three dimensional structure largely determines its function. The process for identifying these structurally similar proteins is called fold recognition and forms the basis of the fold recognition problem.**

Several experimental methods including X-ray crystallography, NMR spectroscopy, and electron microscopy have been used to determine protein structure. However, due to the significant cost and time for using those methods, the number of proteins with known structure(s) is significantly smaller than the number of known protein sequences (i.e. by a factor of approximately 200) and the sequence-structure gap is still increasing.[7]

This project aims at training neural networks to predict if a given query-template protein pair belongs to the same structural fold. Thus trying to achieve accurate comparison of a protein pair.

***Keywords*—Neural Networks, Protein Fold, Protein Folding, Protein Fold Recognition**

I. Introduction

Several experimental methods including X-ray crystallography, NMR spectroscopy, and electron microscopy have been used to determine protein structure. However, due to the significant cost and time for using those methods, the number of proteins with known structure(s) is significantly smaller than the number of known protein sequences (i.e. by a factor of approximately 200) and the sequence-structure gap is still increasing[7]. Therefore, the construction of computational approaches and tools to predict a proteins three dimensional structure from its sequence is an important problem not only for understanding the relationship between protein structure function and advancement in protein-based biotechnologies and drug discovery[10].

An important task in protein structure prediction is to identify proteins that have similar tertiary structures (from among those that have already been determined experimentally). By identifying such proteins, their structures can be used as a template to model the unknown structure of another protein[7].

II. Scope

We construct a neural network to predict if a given query-template protein pair belongs to the same structural fold. The dataset used for training the neural networks is constructed such that protein pairs have lower than 40 percentage sequence identity. A protein from the dataset is paired with all the other proteins and pairwise similarity is calculated. Pairwise similarity features are obtained through five types of sequence alignment and/or protein structure prediction tools (i.e., sequence-sequence alignment, sequence-family information, sequence-profile alignment, profile-profile alignment and structural information) and selected based on their use in prior works[7]. These pairwise similarities serve as inputs to the neural network. 84 such features are considered.[7]The neural network is trained and tested with a dataset file of size 3.8MB[7].

III. Implementation

A. Input Module

The protein dataset consists of proteins with less than 40% similarity which is extracted from the SCOP dataset as per prior works[7]. Pairs of proteins from this database are formed by combinations of one protein with every other protein. These pairs are then used to produce the input to the neural network.

Pariwise input features in categories of sequences, protein families, sequence-sequence alignment, sequence-profile alignment, and profile-profile alignment[7]. These input values are scores generated from number of external alignment tools including MUSCLE, CLUSTALW, T-Coffee, HHSearch, HMMer, BLAST, PSI-BLAST, IMPALA, PALIGN, PRC, and Compass[7].

84 feature comparison results of each protein pair constitutes 1 row in the dataset. The last number on each row is a 1 or a 2. 1 signifies that the two proteins are similar and 2 signifies that they are not similar. We have taken 1,2 and not 1,0 as the class labels as we are using these class labels directly in the formula of the "normalise dataset function. If we take a 0 then there occurs a "divide-by-zero-error". So instead of converting the class labels from 0,1 to a set which does not cause an error,we are directly using 1,2 as the class labels. It doesnt reduce the size of the program by a large factor. But it

does reduce and as we know that every little drops forms an ocean. It is an innovative method that we have used here. We have also used the concepts of k-folds where the training test set is alternatively selected from the dataset provided to it for training.

B. Neural Network implementation module

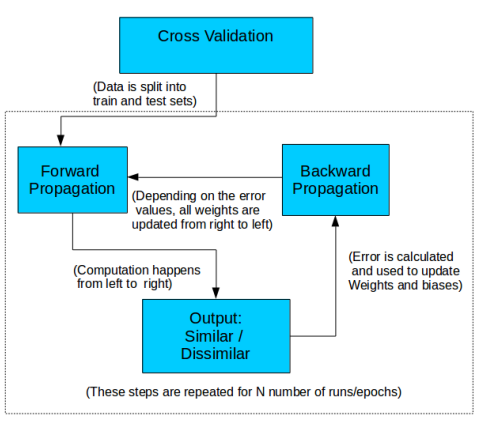

Fig. 1. Block diagram of Neural Implementation

Cross Validation is performed which means the dataset is divided into training and testing sets. For instance there are 10 comparison scores in the dataset, the first 9 scores will be used as training set and the 10th score will be used as the testing set. For the next iteration, the first 8 scores and the 10th score will be used as the training set and the 9th score will be used as the testing set and so on. Thus the entire dataset is trained as well as tested. The neural network is fed scores and the forward propagation computes whether the proteins are similar or not by giving the output as 1 or 2.

Depending on the output obtained, error is calculated. If the actual output is equal to the desired output, there is no error. But if it does not matchwith the desired output then there is an error.

After calculating the error, we activate the back propagation function where the error value is used to update all weights in the neural network. After updating the weights, forward propagation function is activated and it again computes the result.

These steps are repeated for N iterations which is specified in the program.

IV. ALGORITHM

The Algorithm used in this project : Recurrent Neural Networks Our system consists of 3 layers. Input layer, hidden layer and output layer. The input layer consists of 84 nodes as we have 84 comaprison results to be considered.

Explanation of backpropagation with respect to our system:

Sigmoidal function is implemented by this algorithm. A binary sigmoidal function has binary 0 and 1 but class labels used are 1 and 2, hence it is deduced that this algorithm is using a sigmoidal function. The use of the sigmoidal function will be clearer as we explain further. The learning rate is a

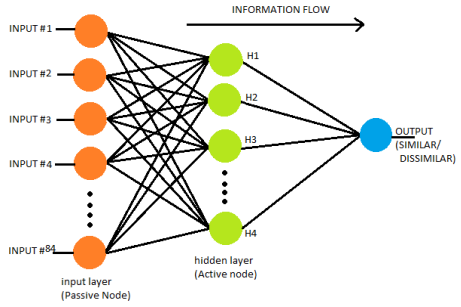

Fig. 2. Block diagram of Neural Implementation

variable value which can be changed manually. In our system, we have kept it as 0.3. The use of the learning rate will also be clearly understood .

Initially random weights are assigned to each synapse(connection between nodes of different layers) in the system. Based on these weights and the values for each of the 84 nodes, an output for each of the nodes is calculated by multiplying the weights and the node values.

The input to the next node is calculated by applying the sigmoidal function to the output of the node to which it is connected. This is called forward propagation. The reason why we use the sigmoidal function is that the derivative of the sigmoidal function can be calculated by simple multiplication and hence no need to use calculus libraries.

If f(x) is the sigmoidal function then its derivative is defined as f(x)*[1-f(x)]. This is how simple it is. Moving on, by doing this, we can find out the input to the output layer and hence its activation function. The activation function is nothing but the sigmoidal function.

Now based on the sigmoidal function of the output layer, we calculate the error in the final output. This is done with the help of stochastic gradient descent. It is a mathematical concept, where we try to achieve a global minima.

Global minima in terms of error is achieved in our case because we want to minimise the error. The number or amount of steps on the graph which we need to take in order to achieve global minima are giverned by the learning rate.

If we take a very high learning rate then the we might miss out on reaching the global minima because then we would be varying the weights by a large amount. Also if we take a very small learning rate then it would take a lot of time to achieve the global minima. Hence we selected the learning rate in such a way that it is optimum. Based on this error calculated, we calculate the change in weights for all the synapses. This change in weights is then added to the original weights and the new weights are found. This is called back propagation of error.

This goes on till the number of epochs is equal to the number of epochs that we have defined. Finally when all the epochs are done, the final weights are found. Based on these weights, the system can predict an unseen pair to be similar or dissimilar.

V. Conclusion

In this work, we have constructed a neural network such that given a protein pair it tells us whether it belongs to the same fold The accuracy that the system varies with the size of the dataset. With a smaller dataset of the order of Kbs, it can give an accuracy of 100%. But as the dataset size increases, goes in the orders of Mbs, it gives an average accuracy of around 75%. Also, the accuracy depends on the learning rate. Very high or very low learning rate may reduce the accuracy. Hence for every dataset(size) we take, we need to tweak the learning rate so as to achieve a high accuracy. By training and getting an experience of a number of datasets having different sizes, we could get used to a constant or a small range of learning rates. For the current testings that we did, we found a learning rate of 0.3 to be optimum. Moreover,the accuracy also depends on the number of folds and number of epochs taken. More the number of epochs and folds, greater will be the accuracy but the time taken to give the output will increase. Please note that the accuracy of the system doesnt take into account the time factor. It just considers the correctness of the predictions. Hence in order to keep a balance between the time taken and correctness, we need to take optimum number of epochs and folds. As the paper contains interdisciplinary work the literature survey took quite a while.

Also bioinformatics offers an abundance of data on their various websites like PDB, UniProt, NCBI etc. Streamlining the data which we would be working with so as to provide a notable contribution to the fold recognition was a tedious procedure. Data Preprocessing module was undertaken and thus working on feature generation scripts for obtaining scores from the output files was a task that was taking a long time. After researching a bit more we could extract datasets from a repository collated by Taeho Jo, Jie Hou, Jesse Eickholt Jianlin Cheng[7].

This paper considers a basic Recurrent Neural Network and tells if the given two poteins have similar folds. Further scope can be increased by adding a Data preprocessing module. This would involve extracting input features by giving the protein pairs present in their FASTA format to various sequence alignment tools and applying feature generation scripts to extract the scores from the output files. Also the neural network module can be replaced with modules using more complex algorithms like HMM algorithm, Restricted Boltzman machines, Convolutional Neural Networks etc to offer comparisons on which handles the Protein Fold Recognition problem better.

Acknowledgment

This paper would not have been possible without the kind support and help of many individuals and Organizations.We would like to extend my sincere thanks to all of them. We are highly indebted to prof. Uday Nayak for their guidance and constant supervision as well for providing necessary information regarding the project and also for his support for completing the Project.

We would like to express my gratitude towards our project Coordinator Prof. Tayyabali Sayyad, Head of The Department IT, Prof. Janhavi Baikerikar and Principal of DBIT, Dr. Prasanna Nambiar for their kind co-operation and encouragement which helped us in completion of this project.Our thanks and appreciation also go to our parents and the entire college staff who willingly helped us out with their abilities.

References

[1] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, Philip E. Bourne; *The Protein Data Bank. Nucleic Acids Res* 2000; 28 (1): 235-242. doi: 10.1093/nar/28.1.235

[2] Leyi Wei and Quan Zou;*Recent Progress in Machine Learning-Based Methods for Protein Fold Recognition*;School of Computer Science and Technology, Tianjin University, Tianjin 300354

[3] Jianlin Cheng, Pierre Baldi; *A machine learning information retrieval approach to protein fold recognition*; *Bioinformatics* 2006; 22(12):1456-1463.doi:10.1093/bioinformatics/btl102

[4] Stephen F. Altschul, Thomas L. Madden, Alejandro A. Schffer, Jinghui Zhang, Zheng Zhang, Webb Miller, David J. Lipman; *BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res* 1997; 25 (17): 3389-3402. doi: 10.1093/nar/25.17.3389

[5] BLAST http://en.wikipedia.org/wiki/BLAST , last modified on 12 April 2017, at 02:26

[6] Jo, T. et al. *Homology Modeling of an Algal Membrane Protein, Heterosigma Akashiwo Na+ ATPase. Membrane* 35(2), 8085 (2010).

[7] Taeho Jo, Jie Hou, Jesse Eickholt Jianlin Cheng; *Improving Protein Fold Recognition by Deep Learning Networks*; SCIENTIFIC REPORT 5, ARTICLE NUMBER: 17573 (2015)

[8] Baker, D. Centenary Award and Sir Frederick Gowland Hopkins Memorial Lecture. *Biochemical Society Transactions* 42(2),225229 (2014).

[9] Pressman, Roger (2010). *Software Engineering: A Practitioner's Approach. Boston: McGraw Hill*. pp.4142. ISBN9780073375977.

[10] Murzin, A. G., Brenner, S. E., Hubbard, T. Chothia, C. *SCOP: a structural classification of proteins database for the investigation of sequences and structures.* Journal of molecular biology 247, 536540 (1995).

[11] Matt Spencer, Jesse Eickholt, and Jianlin Cheng; *A Deep Learning Network Approach to ab initio*;IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, VOL.12, NO. 1, JANUARY/FEBRUARY 2015

[12] Timothy K. Lee,Tuan Nguyen;*Protein Family Classification with Neural Networks*, STANFORD UNIVERSITY

[13] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, Philip E. Bourne; *The Protein Data Bank. Nucleic Acids Res* 2000; 28 (1): 235-242. doi: 10.1093/nar/28.1.235

[14] Leyi Wei and Quan Zou;*Recent Progress in Machine Learning-Based Methods for Protein Fold Recognition*;School of Computer Science and Technology, Tianjin University, Tianjin 300354