The usage of machine learning paradigms on protein secondary structure prediction

Hanan Hendy, Wael Khalifa, Mohamed Roushdy, Abdel-Badeeh M. Salem
Computer Science Department
Faculty of Computer and Information Sciences
Ain-Shams University
Cairo, Egypt
hanan.hendy@cis.asu.edu.eg, wael.khalifa@cis.asu.edu.eg, mroushdy@cis.asu.edu.eg, absalem@cis.asu.edu.eg

Abstract: - The significance of the secondary structure prediction process is something no one can deny. This is because of the importance of protein in all our human system functionalities. Protein forms every single element in the body using its amino acids. These amino acids start to bond together forming other protein structures. A lot of diseases can be diagnosed by simply checking the deformation of these structures. The problem is that it takes a lot of effort to get from the primary protein structure –aka amino sequence– to the secondary, tertiary and quaternary structures it forms. Through the past decade a lot of machine learning methods arose that predicted the secondary structure and then predicted the tertiary from it. Most of these methods were based on Neural Networks paradigm only. This paper aims to show how other machine learning techniques have been used to predict the secondary structure. The techniques used are; Case Based Reasoning, Bayes Network, Decision Tables and Decision trees. The highest accuracy reached was when using Bayes network to predict Beta secondary structure only, it reached an accuracy of 75.89 %.

Key-Words: Case Based Reasoning, Decision Trees, Bioinformatics, Machine Learning, Protein Secondary Structure Prediction.

1 Introduction

The process of protein structures prediction aims to predict the different structures with only knowing its previous structure. This means predicting the quaternary structure from the tertiary, the tertiary from the secondary and the secondary from the primary. The aim of secondary structure prediction accordingly is to predict the secondary structures (alpha, beta and coil) given only the primary structure. This primary structure is formed from different amino acids shown in Figure 1. If secondary structure prediction process resulted in accurate findings, this can dramatically help in both disease diagnosis and tertiary structure prediction accuracy as well. One of the challenges that faces the secondary and tertiary structures prediction is the small number of secondary and tertiary known structures compared to primary structures. According to 2011 last survey; there were only 70,000 known tertiary structures in the Protein Data Bank (PDB) [2] compared to 12.5 million protein sequences in the RefSeq database [3].

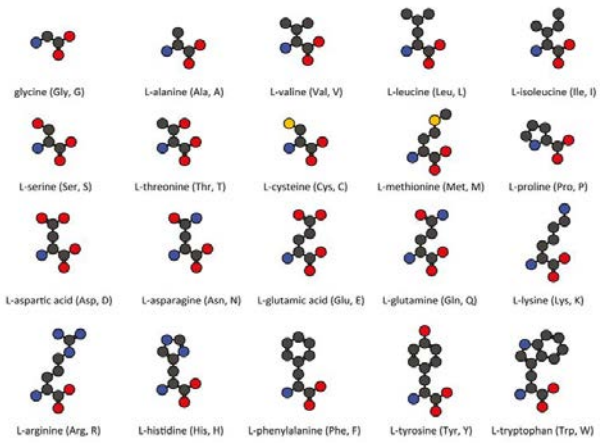

Fig. 1 - Amino Acids [1]

It is important to know how the secondary structure is formed before getting into knowing the techniques used in the prediction process. When the amino acids bond together forming hydrogen bonds this results in one of the three commonly known secondary structures. They form either Alpha Helix, Beta Strands/Sheets or Coils as shown in figure 2.

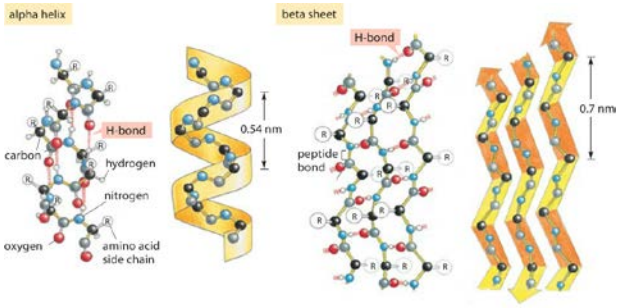

Fig. 2- Hydrogen bonding and protein secondary structure [4]

The secondary structure prediction started long ago even before machine learning techniques were discovered. It started with pure statistical methods with Chou-Fasman and GOR methods as shown in [5]. Later machine learning techniques started to get involved in the prediction process but most of them focused only on Neural Networks. The accuracy of these methods ranged from 50% to 85%. Janice Glasgow et al. [6] used case-based reasoning approach and reached an accuracy ranging from 65.40% to ~83%.

In this paper, section one presents an introduction about the protein secondary structure prediction, the techniques used and. Section II shows the data pre-processing. Later, section III demonstrates the implementation of case-based reasoning, decision tables and trees showing the results obtained. Finally, the conclusion is presented showing what future enhancements can be done and which experiments to be conducted to increase the accuracy.

2 Data Pre-Processing

Protein primary structures vary from one to another in the length, this is considered one of the challenges that needs to be solved before thinking of the prediction process. Another challenge is dealing with separate files to get out a single file to be used in the prediction process.

All the experiments conducted in this paper used a set of protein extracted from the PDB [2]. The database named CB513 [7] is the one mainly used. It has 513 different protein sequences with their secondary structures. The CB513 proteins' length vary from 20 amino acids to 754 amino acid with average of 164 per structure. The dataset comes in separate files each encoded in the FASTA format [8]. Each file has data about singe structure, what we are interested in is only the primary and the secondary structures. The output of the pre-processing step is having a unified (same length), collected (all proteins together) dataset to start the prediction process.

The preprocessing passes by multiple phases, starting with dealing with the raw files, passing through the encoding phase then getting to combine the data in the format needed for the prediction. Two results formats are needed; one is CSV format and the other is a CSV-like format used in WEKA [9] classifier. Figure 3 sums up the stages that will be discussed later.

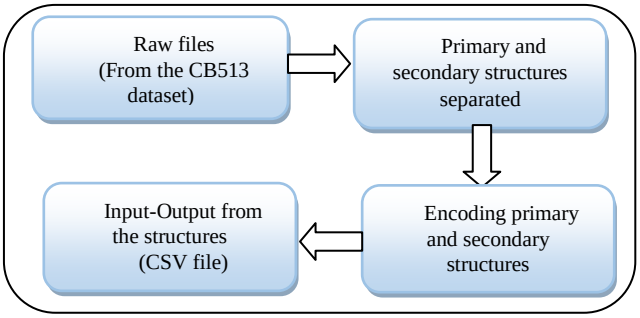

Fig. 3 - Block diagram for protein data preparation

2.1 Dealing with raw files

The FASTA format [8] is a text-based format for representing either nucleotide sequences or peptide sequences, in which nucleotides or amino acids are represented using single-letter codes. The only information needed from all the data in the extracted file is the primary sequence and the secondary sequence. Both are found in the file with the headers RES, DSSP respectively.

2.2 Primary and secondary structure separation

The first step done on these files is collecting all the primary structures and the secondary structures together in two separate files removing all commas and replacing the '_' with a C (coil).

2.3 Encoding primary and secondary structures

The next step after having the primary and secondary structures separated is to encode them. The encoding step includes assigning a numeric value for each letter of the input and output. First each of the amino acids in the primary structure is represented as a number from 1-20. Then each of the secondary structures are repressed as a number from 1-3 (alpha, beta and coil). After finishing this step, two files are there one having all encoded primary structures and the other having all encoded secondary structures as explained above.

2.4 Preparing input and output as CSV files

After encoding the primary and secondary structures it is time to forma them to start the learning process. The first thing done is to unify the lengths. Since they have different lengths, the idea is to generalize the problem in which the problem is seen as inputting a fixed length sequence and deciding whether it's alpha helix, beta sheet or coil. How will this be done? By choosing a number then inputting amino acids with this length and predicting the mid amino acid. This means that as if the prediction is for the centered amino acid with respect to the neighbors. The corresponding output will be a number representing alpha helix, beta sheet and coil at the mid index of the chosen number. This number is constant per experiment. For example, choose the number 7 which means that the first 7 amino acids will be the input and the output will be the secondary structure mapped to index 4 (as if the amino acid number 4 is being checked when it's in a sequence of 7 amino acids). To be more specific, the goal is to learn the way the N amino acid –here 7– will interact and bond producing the secondary structure at N/2 position.

To achieve this, all primary sequences are combined together with "00" separator also the same is done for the secondary sequences. This was we have a very long single sequence to predict. Later this primary sequence is divided with the value chosen before (i.e. 7) by moving this window one step each time. When a “00” appear at the mid position the current window is discarded as it’s the mapping of the concatenation and not a real amino acids bonding as shown in this example:

*151703**02**2006121812*150803171706011602091701091 30
*151703022**0**06121812150803171706011602091701091 30*
*1517030220**06**12181215080317170601160209170109130*

Then the output bond for each is taken by skipping the first 3 numbers as the first to have a secondary output will be at the 4th position.

This way the unified lengths sequences are ready. They are then appended in a file to match the CSV format (for myCBR tool) and CSV-like -.arff- (for WEKA) as shown in the below example in Figure 4.

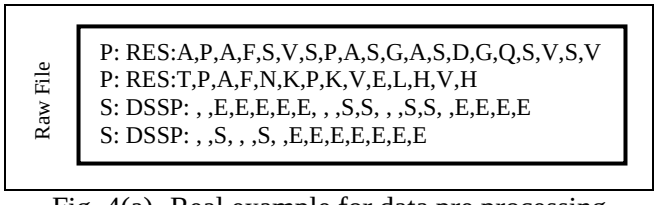

Fig. 4(a) -Real example for data pre processing P and S refers to primary and secondary structure files respectively.

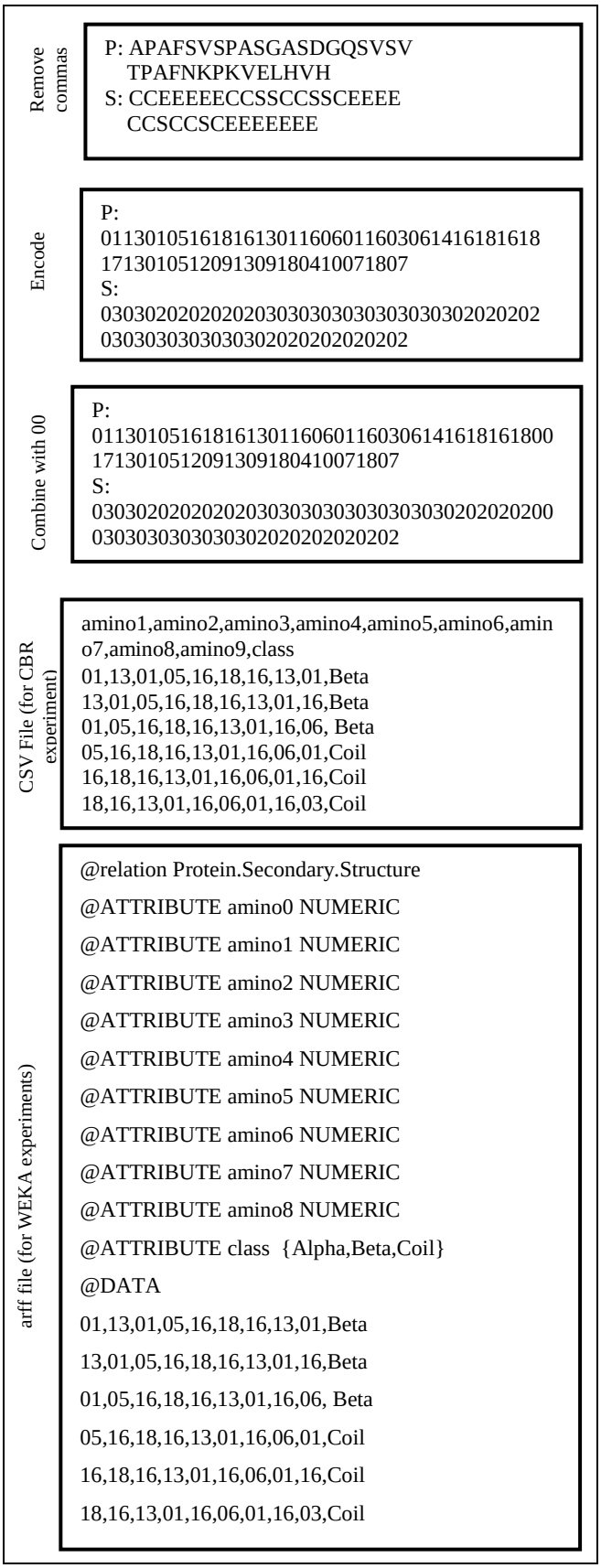

Fig. 4(b) - Real example for data pre processing P and S refers to primary and secondary structure files respectively

3 Implementation and results

Multiple machine learning techniques have been tested. This section goes through the different experiments and discusses their accuracy. These methods were used in prediction but not for protein structures. Some predictions were conducted on Breast Cancer Recurrence as discussed by Siddhant Kulkarni and Mangesh Bhagwat [10]. Also, S. Venkata Lakshmi and T. Edwin Prabakaran disused using WEKA for intrusion detection [11]. The accuracy is discussed with choosing the constant as 17, 19, 25 and 31. Moreover, the experiments were conducted once to predict alpha, beta and coil all together and other times predicting alpha only or beta only.

WEKA version 3.6.13 [12] is used for experimenting decision tables, decision trees and Bayes network. The evaluation of these methods was discussed by Chitra Nasa and Suman[13]. For this experiment purpose, the data is divided as 66% training and 34% testing.

3.1 ZeroR

The first conducted experiment is the ZeroR [14] in which no prediction is done. It's considered the simplest classifier. It classifies simply by choosing the majority class. The reason behind choosing to start with ZeroR is to put a threshold for any other predictor as if the accuracy is less than the ZeroR then it is not meaningful. Tables 1 sums the results of the ZeroR predictor predicting both alpha and beta then predicting each alone.

Table 1 - Prediction Accuracy (ZeroR)

Both alpha & Beta				
Constant	17	19	25	31
Accuracy	43.9622%	44.1243%	44.1227%	44.1223%
Alpha only				
Constant	17	19	25	31
Accuracy	56.1661%	56.3464%	55.8271%	55.8631%
Beta Only				
Constant	17	19	25	31
Accuracy	67.1051%	67.3832%	67.5846%	66.5936%

3.2 Bayes Network

The second experiment was done using Bayes network [15]. A Bayesian network is a probabilistic directed acyclic graphical model that represents a set of random variables and their conditional dependencies via a directed acyclic graph. The computation parameters were:

- Estimator algorithm: Simple estimator (Estimates probabilities directly from data)
- Search algorithm: Local search K2

Table 2 summaries the predication accuracy of the Bayes network. It sums the accuracy of prediction both alpha and beta then predicting each alone. The accuracy of predicting both alpha and beta ranged within the 62%. On the other hand predicting each alone showed enhancement in the accuracy. Beta prediction showed better accuracy of around 75% which exceeds alpha prediction by almost 2%.

Table 2 - Prediction Accuracy (Bayes Network)

Both alpha & Beta				
Constant	17	19	25	31
Accuracy	62.3431%	62.4487%	62.634%	62.4398%
Alpha only				
Constant	17	19	25	31
Accuracy	73.8694%	74.0637%	74.328%	74.0413%
Beta Only				
Constant	17	19	25	31
Accuracy	**75.891%**	75.4251%	75.6911%	75.4064%

3.3 Decision Table

The third experiment was done using decision tables [16]. A decision table tries to find out if-then rules of a complex model. In our case it tries to find out similarities for amino acids sequences and generate the rules accordingly. The computation parameters were:

- Search algorithm: Best Fit
- Search direction: forward
- Number of non-improving nodes to search: 5

Table 3 summaries the predication accuracy of the decision table. It sums the accuracy of prediction both alpha and beta then predicting each alone. The accuracy of predicting both alpha and beta ranged within the 53%. On the other hand predicting each alone showed enhancement in the accuracy. Beta prediction showed better accuracy of around 71% which exceeds alpha prediction by almost 4%.

Table 3 - Prediction Accuracy (Decision Table)

Both alpha & Beta				
Constant	17	19	25	31
Accuracy	53.8241%	53.9169%	53.8822%	53.4977%
Alpha only				
Constant	17	19	25	31
Accuracy	67.6277%	67.3331%	66.797%	67.2162%
Beta Only				
Constant	17	19	25	31
Accuracy	71.3988%	71.34%	**71.793%**	71.4599%

3.4 Decision Tree

The fourth experiment was done using C4.5 decision trees [17]. A decision tree forms a graph a flowchart like. It tries to find out the relations

between amino acids sequences that cause their bonding. The computation parameters were:

- Minimum number of instances per leaf: 100
- Confidence factor: 0.1

Table 4 summaries the predication accuracy of the decision trees. It sums the accuracy of prediction both alpha and beta then predicting each alone. The accuracy of predicting both alpha and beta ranged within the 53%. On the other hand predicting each alone showed enhancement in the accuracy. Beta prediction showed better accuracy of around 70% which exceeds alpha prediction by almost 6%.

Table 4 - Prediction Accuracy (Decision Tree J48)

Both alpha & Beta				
Constant	17	19	25	31
Accuracy	52.7365%	52.7209%	53.0358%	52.6163%
Alpha only				
Constant	17	19	25	31
Accuracy	66.8236%	64.552%	64.8154%	64.4523%
Beta Only				
Constant	17	19	25	31
Accuracy	67.0142%	70.3775%	**70.413%**	70.1925%

3.5 Case Based Reasoning

All the previous experiments were conducted using Weka tool. This case based [18] experiment was conducted using myCBR tool [19]. The used version of the workbench is 3 and the sdk version 3.1. The data was divided as 70% for training and 30% for testing. The computation parameters were:

- Primary structure length: 17
- Prediction : predict only alpha and beta or predict all the secondary structures,
- Number of cases retrieved: 3, 5 or 10
- Voting of matched cases: Or-ing for the matched cases or accumulated voting of the similar cases.
- Similarity measure: weighted sum of the matched amino acids in their same position.

Table 5 summaries the predication accuracy of the CBR experiment. It sums the accuracy of prediction alpha and beta then predicting all (alpha, beta and coil). Also it shows the difference between training with cases that contain alpha and beta only verses training will the full dataset. The accuracy shows that training with alpha and beta only is better than training with the full data set. This is because the coil secondary structure is dominant and affects the similarity, so it's seen as similar with mostly any of the tested cases. Moreover, selecting 5 similar cases showed the best voting accuracy that reached ~62%. The last experiment in which oring the selected cases and if any of them matched the expected output it's counted correct, showed an accuracy of ~89%.

Table 5 - Prediction Accuracy (CBR)

Train with alpha, beta and coil					
Number of retrieved cases.	10		5 (voting)		5 (or-ing)
Accuracy	46.4%		45.58%		**88.7%**
Train with alpha and beta					
	Test with alpha, beta and coil	Test with alpha and beta	Test with alpha, beta and coil	Test with alpha and beta	Test with alpha and beta
Number of retrieved cases.	3	3	5	5	10
Accuracy	33.1%	60.5%	33.8%	**61.9%**	55.2%

By conducting the above experiments the following conclusions were made:

- Separating the prediction of alpha and beta secondary structures is better than predicting both together
- Beta structure prediction through all the experiments shows the highest accuracy with all varying parameters.
- Using different window sizes didn't show big variation in accuracy.
- Using decision tables and decision trees can help in improving the prediction accuracy.
- Case based reasoning is not the best technique that can be used for prediction but it can be used with different similarity measure and voting technique that may lead to a better performance.

4 Conclusion

As discussed all through the paper, multiple machine learning techniques were used to predict the protein secondary structure. The paper discussed the accuracy of five different techniques. It can be concluded that other techniques other than artificial neural networks can be used to predict the secondary structures. Although the accuracy didn't exceed those of neural networks but they can be combined together to enhance the accuracy.

Reaching an accuracy of around 75% can be an initial trial to using different techniques in the prediction process. Some enhancements and future work can be listed as follows:

- Combining these techniques and using voting idea to reach higher accuracy.

- Using one of these algorithms as pre-technique to artificial neural networks. Which means testing the amino acid sequence among the above techniques, if they concluded the same secondary structure then this is the prediction else then go for neural networks.
- Try different computational parameters such as changing the search algorithm or the similarity measure in case of CBR.

References:

[1] Protein Structure Study Guide "http://www.alyvea.com/biologystudyguides/protein-structure.php" [Accessed: February 2016].
[2] Rcsb protein data bank." http://www.rcsb.org/pdb/home/home.do. [Accessed: July - 2015].
[3] A. E. Kister, Protein Supersecondary Structures, Humana Press, 2013.
[4] Ron Milo and Rob Phillips "Cell Biology by the Numbers" Ch. 3 July 2015.
[5] Hanan Hendy, Wael Khalifa, Mohamed Roushdy and Abdel Badeeh Salem "A Study of Intelligent Techniques for Protein Secondary Structure Prediction" International Journal "Information Models and Analyses" Volume 4, Number 1, p.p. 3-12 2015.
[6] Janice Glasgow, Tony Kuo and Jim Davies "Protein structure from contact maps: A case-based reasoning approach" Information Systems Frontiers, vol. 8, no. 1, pp. 29-36, February 2006
[7] "Cuff and barton data set." http://comp.chem.nottingham.ac.uk/disspred/datasets/CB513[Accessed: July - 2015].
[8] fFasta format." https://en.wikipedia.org/wiki/FASTA_format. [Accessed: June - 2015].
[9] Ian H. Witten, Eibe Frank and Mark A. Hall. "Data Mining: Practical machine learning tools and techniques" Morgan Kaufmann 2011
[10] Kulkarni, Siddhant, and Mangesh Bhagwat. "Predicting Breast Cancer Recurrence using Data Mining Techniques." International Journal of Computer Applications 122.23 (2015)
[11] Lakshmi, S. Venkata, and T. Edwin Prabakaran. "Performance Analysis of Multiple Classifiers on KDD Cup Dataset using WEKA Tool." Indian Journal of Science and Technology 8.17 (2015)
[12] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.
[13] Nasa, Chitra. "Evaluation of different classification techniques for web data."International Journal of Computer Applications 52.9 (2012).
[14] "ZeroR" http://www.saedsayad.com/zeror.htm [Accessed: February 2016]
[15] Ben-Gal, Irad. "Bayesian networks." Encyclopedia of statistics in quality and reliability (2007).
[16] "Decision Table" https://en.wikipedia.org/wiki/Decision_table [Accessed: February 2016]
[17] "Decision Trees" https://en.wikipedia.org/wiki/Decision_tree_learning [Accessed: February 2016]
[18] Poole, David L., and Alan K. Mackworth. Artificial Intelligence: foundations of computational agents. Cambridge University Press, 2010. Chapter 7.6
[19] Kerstin Bach, Christian Severin Sauer, Klaus-Dieter Althoff, and Thomas Roth-Berghofer: Knowledge Modeling with the Open Source Tool myCBR, Proceedings of the 10th Workshop on Knowledge Engineering and Software Engineering (KESE10, located at 21st European Conference on Artificial Intelligence) , August 2014, CEUR