Prediction of Protein Secondary Structure using Logistic Regression, K-Means Clustering, and Naïve Bayes Variation

Abhishek Sethi1

1 - University of Maryland: Baltimore County  
abhishek9691@gmail.com

Abstract

This paper discusses the use of machine learning applied to a biological setting. The specific case is the prediction of secondary structure for a protein. This is a problem in biology because of the central dogma of proteins that form follows function. If this is true, then that means if we have a strong understanding of the form, then we can have a good idea of the function of the protein. In order to do this, we choose to use the probabilistic capabilities of the logistic regression sigmoid, and then the clustering of k-means to obtain secondary structure information. This paper primarily introduces the method as a way to use commonly known and efficient algorithms to compute and determine the secondary structure of the protein given its amino acid sequence. First the amino acids were passed through the logistic regression sigmoid, then combined with the k-means clustering algorithm. Overall, the final results looked promising for less proteins with similar secondary structures. However, once variety in size, sequence, and structure was introduced, the method’s accuracy drastically decreased from 84% to about 42%. This decrease is discussed in detail, but overall, a slight miscalculation in the logistic regression attributes to this. Overall, this combination of algorithms helps identify a protein’s secondary structure relatively accurately.

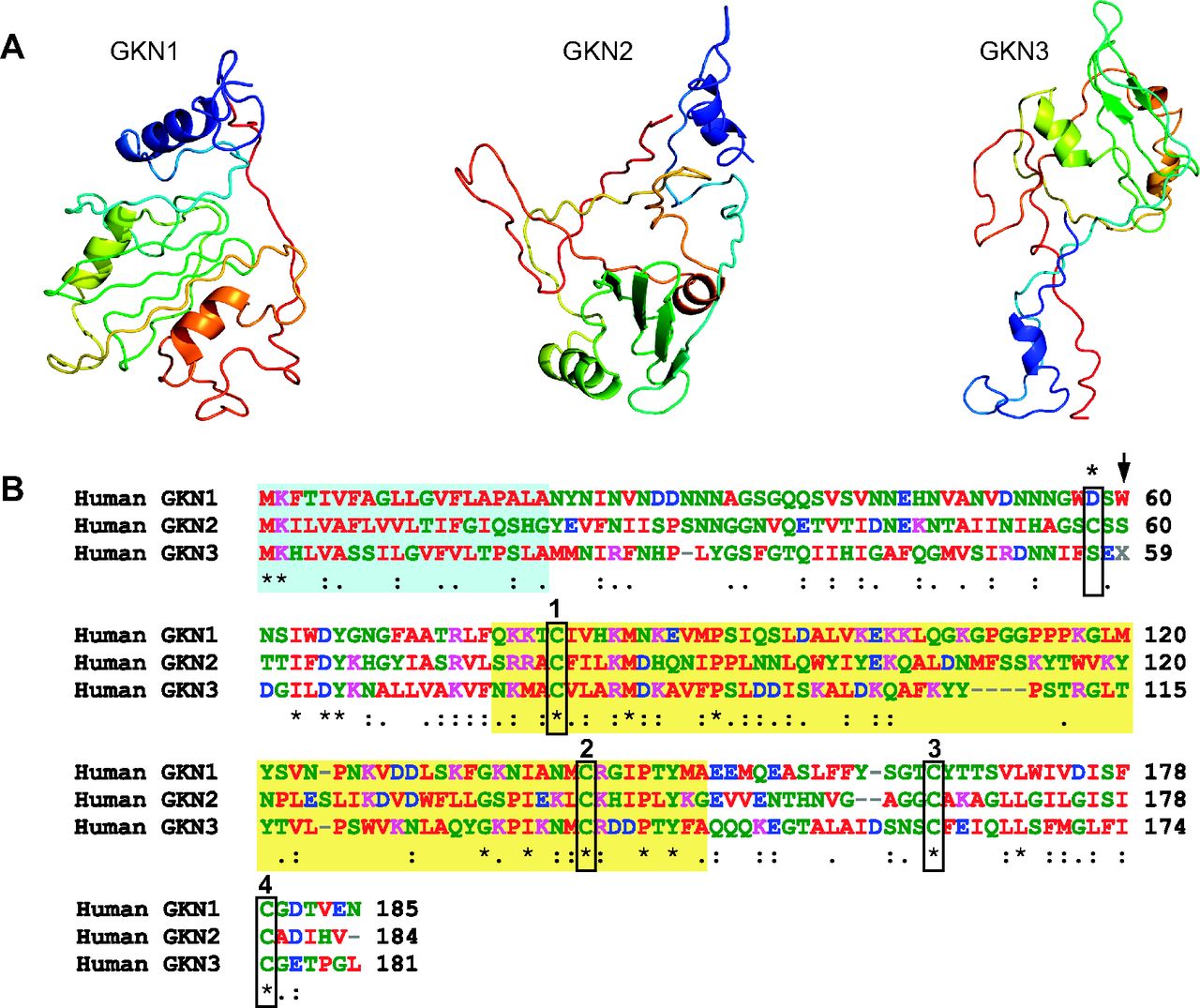
Introduction

Artificial intelligence (AI) and Machine learning (ML) are new and upcoming fields of computer science with limitless applications. ML/AI is when a program can modify some aspect of itself, its state, in order to better understand the surrounding world and produce better outputs when given the same input [6]. An example would be standard reinforcement learning where an agent proceeds through a maze. The first time it is unable to reach the end quickly, but on learning the maze, and adjusting its own parameters, it is able to successfully reach the end of the maze faster and faster. Outside of simple toy examples, ML has also been applied to predicting the occurrence of specific events such as stock market changes, text classification, and so on. This paper will specifically discuss the application of machine learning to a common problem in biology. This problem is predicting the structure of proteins.

Before continuing onwards, a quick background of biology is necessary to better understand the problem. Biology is simply known as the study of life. Life is defined as living matter that shows specific attributes such as growth, metabolism, reproduction, and energy transformation [4]. In order to study these functions, we have to move all the way down to the molecular level. The level of granularity we need to reach is not atomic, where we discuss single atoms such as carbon, nitrogen, etc, but rather we only require reaching the level above that, which is known as the macromolecular level. Macromolecules are made of different atoms as well, but different arrangements of these atoms will produce different macromolecules. For example, a specific organization of carbons, nitrogens, and oxygens will create a DNA molecule, whereas the same atoms organized in another way will create a protein molecule. This is crucial to understanding that this cascades onwards. These atoms are building blocks of macromolecules, which are building blocks of the life functions we mentioned earlier.

The specific macromolecule we will be focusing on is a protein. A protein is a large molecule that plays critical roles in the body, and is required for structure, function, and regulation of bodily tissues [7]. There are hundreds of different kinds of proteins, such as enzymes, antibodies, and structural. These proteins take many different shapes and forms as well. When discussing proteins, it is important to remember that form follows function, meaning however the molecule is shaped gives a hint about what it does in the cell. This just goes to show that knowing a structure of a protein can give important information.

A protein is made up of building blocks called amino acids. These amino acids are strung together through chemical reactions to create the protein’s backbone. Figure 1 provides an example of a folded protein with its backbone below it.



*Figure 1: Part a shows multiple folded proteins and part B shows the amino acid sequence that folded on itself to create the 3-D structure presented above it. [2]*

Once the backbone is completed, the atoms in the amino acids begin to interact with one another, and the backbone begins to fold on itself. Imagine creating origami. Initially, the paper is completely flat, but with many folds, the paper becomes a crane, or frog, or any shape the imagination can create. Protein folding works in a similar manner, but there are many other factors that play a role in protein folding which are outside the realm of this paper. Even though every protein is different, one similarity between all proteins is that the sequences fold up into 1 of 3 different structures, known as an α helix, β sheet, and a coil (not part of a structure). These are all of varying sizes, but the same structure. The one problem that scientists face is determining this secondary structure of the protein sequence. Current methods require a physically solid protein, frozen in ice or some other media, which is hard to do. It is much easier to obtain a protein’s amino acid sequence, and then use computational methods to determine the structure. However, even determining the structure accurately using computational methods is hard to do. Current papers make many assumptions, not accounting for the randomness in nature, but rather an entropy free environment, which almost never occurs. The purpose of this paper is to present a plausible method to determine protein secondary structure by classifying amino acids to one of the 3 secondary structures.

Proposed Method

Many attempts at predicting secondary structure using machine learning techniques use a fixed window, or some assumption that there is no randomness in the data. Holley and Karplus attempted to use a neural network to try and predict protein structure, but also made an assumption that the structures are all of a fixed size of 17, and used that as a sliding window [3]. Support Vector Machines (SVM) have even been shown to create optimal results, achieving almost a 90% accuracy on random data [1]. However, it is worth noting that the researchers only used α helices and β sheets, and that to a small mix or completely one or the other. The level of granularity that would be optimal is to work at the individual amino acid level, and pool the information from all of them together. This is because we want to identify all the separate interacting amino acids, as well as be able to bring together amino acids and see what the overall structure of all of them together would be. In order to do this, the following method is proposed.

The first step is to create features of each protein sequence. The features are simply the amino acids in the protein sequence, since this is ultimately the determining factor when discussing the secondary structure of a protein. There are no other features that could be used for a problem like this unfortunately since we are already working at the finest level for proteins.

**Logistic Regression**

Once we have a protein sequence, our first step is to pass each amino acid through a logistic regression function. The function chosen to use is the standard sigmoid function, shown in equation 1.

S(t) = \frac{1}{1 + e^{-t}}.

*Equation 1: The sigmoid used for logistic regression. t is equal to the weight vector (w) dot product with x (the input vector).*

A total of 3 sigmoids were used, each containing its own unique weight vector. Each sigmoid represented a secondary structure, so each one represented the α helix, β sheet, and coil, respectively. This function helps us provide a probability using the unique weight vector and amino acid for whether or not that amino acid is most likely to be found in an α helix, β sheet, or coil. So in the end, we pass each amino acid through each sigmoid, so each amino acid in the protein sequence will have a total of 3 probabilities assigned to it, one for each secondary structure. The reason for doing this is that because in nature, there are certain amino acids that fit better in certain structures, while other amino acids are almost never found in other structures. The first step was to identify the length of the weight vector. The length was simply established to be as long as the longest protein sequence in the training data. This was done simply so that along with distance and probabilities, we also obtain information in terms of location. If an amino acid is commonly in one location, this implies a trend of the amino acids and how they evolved to be in that specific location of the protein sequence.

In order to perform computation on the amino acid letters with the weight vector, the ASCII value of each was converted to an integer, and that was used in the logistic regression computation. Each structure was represented with a single letter code, A = α helix, B = β sheet, C = coil. However, when performing the computation, if the amino acid was found in the structure we are training, then the dot product between the weight vector and amino acid was weighti \* residue’s ASCII value \* structural code’s ASCII value, where i is the ith weight in the weight vector. For example, if while training the α helix, we find that the amino acid was found to be part of the α helix during training, then we multiply all 3 values out. If not, then we only multiply the first 2 values. The reasoning is because having the amino acid found in that structure for the specific secondary structure should have a big impact on the weight of the weight vector. Other than the transformation from character to integer, the rest of the logistic regression algorithm remained the same. Once probabilities were assigned, the amino acids were then passed onwards to a second algorithm.

**K-Means Clustering**

The next algorithm used on the amino acids after the probabilities were predicted was the k-means clustering algorithm. The standard k-means clustering algorithm was used. There was no specific value chosen for the number of centroids. Instead, a random number of centroids were chosen. This random number was between 7 and 17. Both of these numbers were picked arbitrarily, however, it would have been ideal to use a number in relation to the size of the protein sequence since it is possible that the sequence could have many structures, or only one dominant structure. Once the numbers of centroids were picked, the centroids were randomly assigned from the amino acids inside the protein sequence. From here, the standard k-means clustering algorithm was run. The key thing was how to assign amino acids to each centroid. 2 criteria were used to determine how close an amino acid was to a centroid. The first was physical distance, as in how many residues lie between the centroid and the current amino acid being analyzed. The other metric was the probabilities between the centroid and amino acid. Since we have 3 probabilities for each amino acid, we can determine which secondary structure that amino acid most likely belongs to simply by the maximum of the probabilities. If the α helix probability is the highest, then the amino acid is most likely found in α helices. Using this metric, each amino acid would be compared to the centroid, and would match with the centroid that also has a similar classification. If no such centroid exists, then the amino acid is simply assigned to the first centroid. The probability comparison was used to collect a list of similar centroids, and then the amino acid would select the closest centroid from that list and be assigned to that.

This form of clustering was done until convergence was reached, and then finally the overall classification for the cluster was determined using a Naïve Bayes style probabilistic calculation. Simply by calculating the product of the probability for each secondary structure, and seeing which was the highest ended up determining what the overall classification of that cluster is. Intuitively, this seemed like a good choice because the logistic regression takes care of the fact that each amino acid can belong to certain structures, and then the k-means clustering brings together the amino acids, and then tries to establish what is the dominant structure for that cluster, thus stating which strings of amino acids produce a specific structure.

Experiments

There were 2 phases to the experiments. The first was training of the logistic regression model. Currently, this is done dynamically rather than at one time and done. The training was executed for each logistic regression model using the original dataset. The dataset used for training was obtained from the Protein Data Bank (PDF) [5]. Initially, only 20 proteins were downloaded, and used for training and testing. The main issue with this was that the proteins were structurally similar, as in the same secondary structures in the same places. This resulted in the proposed method to easily recognize the same amino acids, or similar ones, so that the resulting accuracy of the method ended up being 82%. This meant that more proteins were needed.

By simply downloading more protein sequences from the PDB, the search feature allowed for over 300 proteins to be downloaded, all of varying sizes and structures. However, the tricky part was also getting the secondary structural information combined. Unfortunately, this step was done by hand, so that the final format of the input file is shown below:

Sequence: ABCDE…….; Structure: AAAAAAAAAACCCCCCCC…….

Sequence: ABCDE…….; Structure: AAAAAAAAAACCBBBBBB…….

*Figure 2: The structure of the input file*

Note that in figure 2, there is a 1 to 1 correspondence between the sequence and structure strings, so that the length of the 2 is equal. This was the training data used for training the logistic regression model.

Using the trained model, we then ran experiments on the training data set. In order to determine the accuracy and efficiency of the model, once a classification was done for a test data protein, we would then compare the classification by our method to the actual structure that the PDB provided. Accuracy was measured in percentage, and was defined to be how many amino acids were correctly classified by the model out of the total protein length. The experiments involved increasing the training data set, as well as variety of the protein lengths and structures.

Results and Observations

Table 1 shows the results of the experiment

|  |  |
| --- | --- |
| Experiment Type | Average Measured Accuracy |
| 20 protein sequences, similar structure & size | 82% |
| 300 protein sequences, similar structure & size | 74.1% |
| 100 protein sequences, drastically different structures & sizes | 43.7% |
| 300 protein sequences, drastically different structures & sizes | 41.2% |

*Table 1: Shows the results of each experiment that was run. Clearly, there is a trend amongst the accuracy for the experiments.*

As seen in Table 1, it is clear that the method proposed can do very well under the condition that the proteins are structurally similar. This is most likely due to the fact that the logistic regression model labels proteins closely when they are similar in structure. Also, since the weight vector is the size of the largest protein, we end up with the temporal information as well, such as where certain amino acids are located. K-means clustering clearly clusters the similar structures well together when the structures used to train the logistic regression model were similar. It is worth noting that when the structures were similar, and the size of the training set increased, it appears that the accuracy dropped about 8%. This is most likely attributed to the fact that the structures and sequences may be the same size, but the actual amino acids are different between proteins, so introducing that slight variation affected training of the weight vector.

Once variety was introduced to the system with different structures and sizes, the accuracy plummeted. Accuracy decreased from 82% all the way down to 43.7%. This shows that the change in size and structures caused the method to fail. Upon further inspection, this was because the sigmoid ended up being either 0.5 or 1 a majority of the time. This indicates that the weight vectors were not properly trained, or the computation of each amino acid was incorrect. By using the ASCII values, we do not provide any variation between the letters, only a maximum of 26, even though in nature amino acids vary greatly in terms of structure and uses. Rather, it would have been better to have some function that maps the amino acid letter to a scalar that is not the ASCII value, but something else so that variation can be introduced and each amino acid has different influences. Also, the way that points were assigned to a centroid could have also attributed to the decrease in accuracy. By using a strict distance metric, this caused a bias towards closer amino acids, even though it may be the case that further ones are the ones the current amino acid interacts with. Also, since the logistic regression training came out incorrectly, the resulting clustering based on that was incorrect as well. Overall, there were good points of this method, but also a few flaws that hindered its performance.

Conclusion

Overall, this method seemed to have work intuitively and in theory. However, due to the choice of clustering methods, as well as the choice of computation with the amino acids, the implementation resulted in a not so good method. Numerically, it seems worse than the 90% SVM method, but the proposed method also takes into account variety in the structure, as well as extra secondary structures, as opposed to the strict rules that the SVM method enforced. The method proposed using logistic regression combined with k-means clustering is a good method, but can still be refined further to push the accuracy up and possibly be used in further bioinformatics studies.

References

1. Cai, Y. "Prediction of Protein Structural Classes by Support Vector Machines."*Computers & Chemistry*. 26.3 (2002): 293-296. Print.
2. Gastrokines: stomach-specific proteins with putative homeostatic and tumor supporessor roles. Menheniott R. T, Kurklu B., Giraud S. A. American Journal of Physiology – Gastrointestinal and Livery Physiology. Jan 2013, 304 (2) G109-G121; DOI: 10.1152/ajpgi.00374.2012.
3. Holley, LH, and M Karplus. "Protein Secondary Structure Prediction with a Neural Network." *Proceedings of the National Academy of Sciences of the United States of America*. 86.1 (1989): 152-6. Print.
4. "life". Encyclopædia Britannica. Encyclopædia Britannica Online.  
   Encyclopædia Britannica Inc., 2014. Web. 11 Dec. 2014  
   <http://www.britannica.com/EBchecked/topic/340003/life>.
5. Protein Data Bank. http://www.rcsb.org/pdb/home/home.do
6. The Machine Learning Dictionary. http://www.cse.unsw.edu.au/~billw/mldict.html#mclng
7. What are proteins and what do they do?. http://ghr.nlm.nih.gov/handbook/howgeneswork/protein