Protein Sequence Motif Extraction Using Decision Forest

Bernard Chen[1], Cody Hudson[1], Minwoo Kim[1], Aaron Crawford[1], John Wright[1], and Dongsheng Che[2]

[1]University of Central Arkansas, Department of Computer Science, Conway, AR, 72034
[2] East Stroudsburg University, Department of Computer Science, East Stroudsburg, PA, 18301

***Abstract*— As one of the more prominent areas of bioinformatics research, protein sequence analysis has gathered considerable interest. The structure, function, and activities of the protein are strongly linked to structural motifs found in its sequence data. Building off of past research, we propose a new granule model that combines the strength of fuzzy logic and granule computing, with the speed and robustness of a decision tree for the purpose of identifying and extracting protein motif data that transcends protein families. We propose parameters for the model and test their effectiveness using several measures of accuracy and quality. The end result, a decision tree example, is explored for its usefulness in this endeavor.**

***Index Terms*—FGK Model; Decision Forest; Entropy Threshold; Protein Sequence Motif;**

I. Introduction

As one of the basic components of an organic body, proteins have been of prominent interest for many years now in various fields of study. As such, their shape, their functions, and the analysis thereof have become increasingly important. In the past, the process by which one would link both protein structure and shape to its function was through arduous and time consuming methods[1] that included well known processes such as crystallography[2], spectroscopy, and various others. However, in recent years, the promising field of bioinformatics and its accompanying data mining techniques has broken into novel ground by looking not directly at the shape of the protein, but rather at its base composition. Doing so allows the prediction of the three dimensional shape of the protein within an acceptable threshold of accuracy.

To understand this, one must understand that a protein can be described by three basic categories: primary, secondary, and tertiary structure. A protein's primary structure, or "base composition," is its amino acid sequence. These are the building blocks of proteins and the repeating patterns therein are known as motifs. Each of these amino acids can have non-covalent, intermolecular reactions with other amino acids within the protein, causing repeating patterns of folds and sheets within the protein's structure. This localized substructure describes the protein's secondary structure. Finally, the tertiary structure of the protein is the overall three dimensional shape. This is important because not only does the tertiary structure of a protein denote its function, but biochemical research and data would suggest a protein's shape is heavily determined by its primary structure (assuming the absence of any denaturing agents, such as heat or acid)[3]. This supports the idea of using data mining and bioinformatics as a tool for analyzing the primary structure in order to predict the tertiary structure of a protein.

Naturally, in order for analysis of protein data to occur, the data has to be both available and numerous, which suggests that databases are good repository of protein information. Three of the most popular protein databases would include PROSITE[4], PRINTS[5], and BLOCKS[6]. Each describes, in some detail, the various structures of the protein, and, to some degree, also supports the idea that reoccurring primary and secondary structural patterns suggest common tertiary structure.

Various researchers have tried using such databases and numerous techniques[7] to glean some meaningful correlation between protein structure and its three dimensional shape. One such study by Han and Baker utilized their K-means clustering algorithm [8, 9]. Using said algorithm, the protein motifs discovered by it, and an additional algorithm, Hidden Markov Model [10], they were able to predict with some level of success the local tertiary structure of various proteins. In the previous works related to this paper, a Fuzzy C-means algorithm was used to initially break the data into ten subsets. A K-Means algorithm was then utilized to refine each subset. This combination (noted as the FGK Model), was used to not only analyze similarities among protein structures, but also to eliminate low quality data [11]. Support Vector Machines were then proposed to be used for the purpose of predicting the shape of the protein using the above analysis [12].

Granted such, the methodologies proposed within this paper suggest the use of decision trees in the stead of SVM. This method would be used to adequately analyze a protein's primary and secondary structure, as well as offer the ability to use such trees for the prediction of the protein's tertiary structure.

Decision tree algorithms offer output in an easy to understand format, showing precisely how the algorithm made its decisions [13, 14]. Unfortunately, the algorithm requires the calibration of several parameters, including entropy threshold, data classifiers, and labelers. This paper discusses how each parameter has been chosen for further research on the matter.

Therefore, the ID3 (Itemized Dichotomizer 3) decision tree[14] is being proposed to extend the before mentioned previous works related to this paper (the FGK Model)[11]. With its ability to define whether proteins belong to a given cluster, it will be instrumental in eliminating noisy or meaningless data. As the purpose is to discover small, sequential patterns within the amino acid sequence in order to relate to common tertiary structures, it is only natural that not all data will be important. Thus, in this paper, the use of the ID3 decision tree algorithm in order to relate patterns of primary and secondary protein structure to its tertiary structure will be discussed. Just as well, the processes by which its parameters are decided for this particular solution will be heavily discussed primarily through the use of statistical charts describing the output of the decision trees. The following sections of the paper will be arranged as such: methods (describing both present and past approaches used to solve this problem), experimental setup (describing the input in more detail, all utilized equations, etc.), results, future works, and conclusion.

II. METHODS

2.1 Data Set Challenges-Large and Random

As one might suspect, to adequately analyze protein primary sequences, one must overcome the challenges the data presents. The sheer size of the dataset can make even fairly robust data mining techniques seem rather inadequate. Coupled with the inherent random and noisy nature of pulling data from various, somewhat disparate databases [4, 5, 6], the task becomes even more difficult. This is particularly despairing in the case of using a decision tree as it is fairly susceptible to outliers and random data. However, previous works suggest that a preliminary analysis of the data with the "FGK Model"[11], tackles both of the before mentioned problems with a promising level of success. The data can then be further and efficiency processed by the proposed ID3 algorithm.

Granted such, our previous works refers to the experiments of Wei et al [15], which handles, specifically, the randomness aspect of the protein data set. Using the basic idea of the K-Mean clustering algorithm, one will note that all initial centroids are randomly chosen. This potentially renders the algorithm worthless in data that is fairly random in the first place. Instead, they proposed that one run the K-Means algorithm five times. In each round, the randomly generated initial points that had the potential to form clusters with high structural similarity were chosen for the improved K-Means clustering algorithm. These were checked against other potential points, and if its minimum distance fell within a given threshold, it was included as an initial centroid.

The method used in the "FGK Model" was similar, but used a method more akin to averaging the results of the five K-Means runs to produce centroids for a sixth iteration. The resulting clusters from this additional run of the "Greedy K-Means"[11] algorithm used these centroids to produce clusters of various qualities. These qualities are determined by analyzing secondary structural similarity of the proteins in each cluster (the equation for such is given in section 3). Each cluster and its respective centroids are ranked by these structural similarity values, under the safe assumption that centroids that produce higher quality clusters are more desirable.

2.2 Fuzzy Greedy K-Means (FGK) Model

The problem of an overly large and complex dataset is still a prominent issue. Although the five iterations of the traditional K-Means algorithm and then a sixth application of the so-called "Greedy K-Means" algorithm sufficiently handle a great deal of the noise in the data, it is still undesirably inefficient when dealing with the entire data set at one time. However, the proposed FGK Model presents a solution via a simple concept of granular computing. The concept proposes that a divide-and-conquer idea be used to break the original problem into various subsets that can be more easily processed by any given algorithm. In other words, it breaks the original data set into "information granules." [16, 17] Although one might argue that this is simply spreading the running time across various subsets, this isn't true. This is especially important in the case of the K-Means algorithm, which has a running time that increases significantly with a larger dataset.

Therefore, the combination of the "Greedy K-Means" algorithm, and the concept of granular computing produces the FGK Model. The FGK model essentially breaks the protein data set into ten information granules using Fuzzy C-Means. Then performing the five iterations of the traditional K-Means algorithm and the sixth Greedy K-Means run solves both issues with data complexity and size. The resulting output groups the data into ten information granules containing any number of clusters containing any number of proteins.

2.3 Decision Tree Forest Model

Now, we know the FGK Model adequately processes the data, clustering it according to its primary structure into both granules and further into clusters. Even stating so, this model still needs further tools to produce any novel or interesting findings. Thus, the ID3 decision tree algorithm[14] is proposed to further the model. This produces a mechanism that, once trained in the typical fashion, can tell if any given random protein belongs to a cluster with decent prediction accuracy. This is to say that this paper suggests that a "forest" of decision trees is to be created for each cluster in each granule (producing, with the given dataset, a total of 799 decision trees). Each decision tree in the so called forest is trained on the individual clusters' proteins. This would imply that each decision tree will have a basic idea of the inherent sequential patterns (i.e. motifs) within each protein set, such that it can be used to then analyze a given protein's primary sequence. If the decision tree produces a "yes" (the meaning of which, in this particular context, will be explained in the experimental setup section) for that given protein, then this would suggest that the protein has similar characteristics to the homologous proteins within the cluster, including tertiary structural characteristics. A model of such can be seen in Figure 1, combining the elements of the FGK Model and the new Decision Tree Forest Model, to produce a novel approach that takes the analysis power of decision trees and combines it with the data sorting and cleaning power of the FGK-Model.

Thus, the basic concept of the ID3 algorithm will be followed heavily to produce each of the 799 decision trees.

The algorithm, while simple, is fairly robust with large data sets and adequately accurate for this particular task. Granted such, it seems obligatory to note that any future works related to this would make use of much more appropriate decision tree algorithms, as the ID3 algorithm is largely a proof of concept. This is not to say that any results produced by this algorithm are not applicable, but rather that this research team realizes there are more appropriate, albeit more complex, decision tree algorithms to apply.

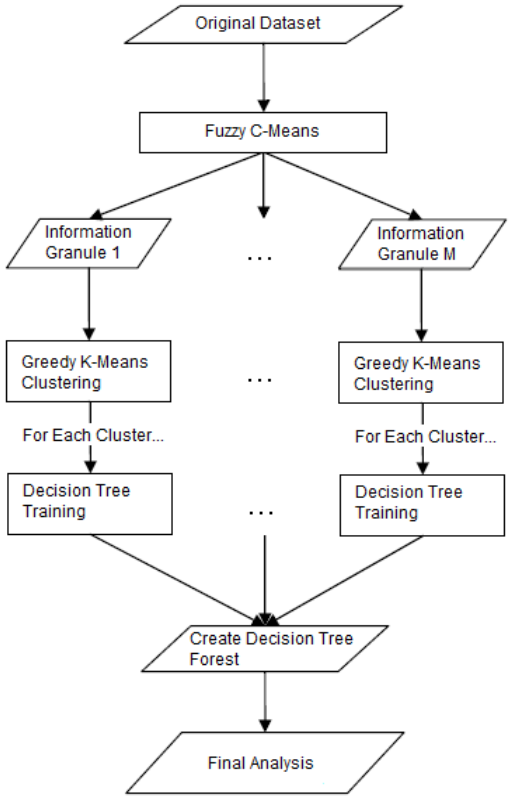

Figure 1. The FGK-Decision Tree Forest Model

III. Experimental Setup

3.1 Dataset

The incoming dataset that is first analyzed by the overlying FGK-Model is composed of 2710 protein sequences obtained from the Protein Sequence Culling Server (PISCES)[18]. None of the protein sequences within this database share more than a 25% sequence identity. Sliding windows with nine successive residues are generated from each protein sequence, such that each window represents one sequence segment of nine continuous positions. Granted such, more than 560,000 segments are generated by this method. Also added to this dataset is the protein's frequency profile, generated from the HSSP[19]. This frequency is based on the alignment of each protein sequence from the Protein Data Bank (PDB), where all the protein sequences are considered homologous in the sequence database. The secondary structure of each protein is also generated from DSSP[20], which is simply a database containing secondary structural assignments for all protein entries in the Protein Data Bank.

The FGK-Model will take this dataset and produce 799 clusters divided among ten information granules. Each granule will have a varying number of clusters within it (this number is determined by a function explained in section 3.4). Each cluster, itself, will have a varying number of protein sequence information in it as well. They will also be of a varying secondary structural similarity (explained in section 3.7). Each of these clusters will then used as the dataset for the induction of each individual decision tree for reasons described in the Methods section.

3.2 Representation of Sequence Segment

As mentioned, the sliding windows of nine successive residues are generated from all of the 2710 protein sequences. Each window corresponds to a sequence segment, which is represented by a nine by twenty matrix, plus an additional nine places corresponding to the secondary structure data obtained from DSSP. Twenty rows represent twenty amino acids and nine columns represent each position of the sliding window. For the frequency profile (HSSP) representations of the protein sequence segments, each position of the matrix represents the frequency for a specified amino acid residue in a sequence position for the multiple sequence alignment. The secondary structure generated from DSSP is simplified from its original eight different classes, down to three. In this paper, structures denoted by H, G, and I are converted to H (Helices), B and E are converted to E (Sheets), and all other structures are converted to C (Coils).

3.3 Distance Measure

As the FGK-Model contains K-Means at its core, a distance formula is imperative. According to various sources[9,15], the most appropriate distance formula to use is the city block metric, as each position in the generated frequency profile will be considered equally. Thus, the following formula is used to calculate the distance between two sequence segments when clustering [9]:

$$\text{Dissimilarity}=\sum_{i=1}^{L}\sum_{j=1}^{N}\left|F_k(i,j)-F_c(i,j)\right|$$

Where L is the window size (in this case nine) and N is twenty, representing the twenty different amino acids. $F_k(i, j)$ is the value of the matrix at row i and column j, which represents, in this case, the sequence segment. $F_c(i, j)$ is the value of the matrix at row i and column j, which represents the centroid of a given sequence cluster.

3.4 FGK-Model Parameter Setup

For the Fuzzy C-Means Clustering that is included in the FGK-Model, the fuzzification factor is set to 1.05 and the number of clusters is set to ten. These settings yielded the best results for this particular dataset. The reason for this being, if the fuzzification factor was to remain constant, but the number of clusters was set to twenty, the membership function would produce nearly equal membership to all clusters for each segment. If one was to decrease the fuzzification factor instead, overflow becomes probable.

In order to separate the information granules generated by the above Fuzzy C-Means results, the membership threshold is

set to twelve percent. Using this value, fifteen percent of the dataset is filtered out and the remaining eighty-five percent is assigned to one or more of the clusters. The formula that dictates how many clusters should be included in each information granule is given below:

$$C_k = \frac{n_k}{\sum_{i=1}^{m} n_i} \times \text{total number of cluster}$$

Where C_k denotes the number of clusters assigned to an information granule k. The number of members belong to information k is denoted as n_k. The number of clusters in FCM is denoted as m. Although using this methodology causes the total data size to increase from 413 MB to 529 MB, as well as an increase in total number of members from 562745 to 721390, it allows for one to deal with one information granule at a time. For example, the largest information granule generated contains 136112 members. From that granule, 151 clusters should be computed from those members, generating a data with the size of 99.9 MB. Compared with the original dataset, the largest granule is only twenty-five percent the size. Therefore, the computation time for all information granules (231720 seconds) is a mere twenty percent of the running time of other leading research [15] (1285928 seconds). These results support the idea that the FGK-Model is a viable one for reducing space and time complexities.

3.5 Decision Tree Induction

The ID3 decision tree algorithm [14], as most classifying algorithms, requires a period of training to produce any level of output. For each cluster generated by the overlying FGK-Model, a decision tree will be trained and generated by considering the frequency profile of each segment in a given cluster. This training produces a resulting decision tree that will now represent the sequential motifs in said cluster. This particular implementation of the ID3 algorithm uses the general formulas for producing both entropy and information gain, both given below:

$$\text{Entropy}(S) = -(S_Y/S_C)\log_2(S_Y/S_C) - (S_N/S_C)\log_2(S_N/S_C)$$

Where S is a collection of total size S_C, S_Y is all items belonging to a given cluster, and S_N is all items not belonging to a given cluster. How these items are labeled is described in section 3.6.

$$\text{Gain}(S, A) = \text{Entropy}(S) - (S_V/S_C)\text{Entropy}(S_V)$$

Where S is a set of each value v of all possible values of attribute A, S_V is the subset of S in which attribute A has the value v, and S_C denotes all items in set S.

3.6 Class Labeling

To generate each label that will determine whether or not a given protein is to be classified as a "yes" protein (that is, it belongs to its cluster generated by the FGK-Model) or a "no" protein, one must consider the secondary structure. For each cluster, a representative secondary structure is generated by determining the secondary structural motif (H, E, or C in this paper) that is most characteristic (that is, the motif with the highest count in that particular column). This is done in each of the nine secondary structural positions for that particular cluster. Once the representative secondary structure for that cluster is generated, each of the proteins are then analyzed for their similarity to this representative structure by both position and the motif at that position and then given an appropriate score. For example, if the representative structure for the cluster (assuming only three structural positions) is HHH, and an individual protein sequence has a secondary structure HEH, then this protein would be given a score of two out of three. For this research, the scores range from 0 (that is, the protein has no similarity to the given representative structure of the cluster) to 9 (which denotes a protein that is fully representative of the cluster). Labeling can then be performed based on this score, such that any values over a certain number, what we will call our label pivot (a parameter discussed in section 3.10), are then considered a "yes" protein. All others would be considered a "no" protein.

3.7 Secondary Structural Similarity Measure

Used in the FGK-Model, the formula to calculate a cluster's secondary structure similarity is given by the following formula:

$$\text{Secondary structural similarity} = \frac{\sum_{i=1}^{ws} \max(p_{i,H}, p_{i,E}, p_{i,C})}{ws}$$

Where 'ws' is the window size and $P_{i,H}$ shows the percentage of helix (H) occurrences among the segments for the cluster in position 'i.' $P_{i,E}$ and $P_{i,C}$ are defined in a similar way in respect to sheets and coils.

Granted such, if the generated structural homology for a given cluster is seventy percent or greater, the cluster can be considered structurally identical [19]. If it falls between sixty percent and seventy percent, it can be said to be weakly structurally homologous [15].

3.8 Average Node Secondary Structural Similarity Measure

Decision trees are defined, primarily, by their nodes, not by clusters. Given such, it is necessary to also include an average node secondary structural similarity measure, given by the following formula:

$$\frac{\sum_{t=1}^{n} |Secondary\_Structural\_Similarity|}{n}$$

Where the "*Secondary_Structural_Similarity*" is the equation defined in section 3.7 and number of decision nodes is denoted as 'n.'

3.9 Ideal Prediction Accuracy Measure

To aid in choosing appropriate parameters, another measure that is made for each decision tree is its ideal prediction accuracy. A twenty-fold cross validation, or similar measure, isn't used in this particular case due to the sheer size of the data as well as the fact that each decision tree is tested on twenty-one different entropy threshold values (described in section 3.10). Instead, the ideal prediction accuracy measure is generated by simply running the training data (that is, the frequency profile of each protein in a given cluster) through

the same tree it produced. This is done by comparing the labels given to the test data by methods explained in section 3.6, against the decisions made by the decision tree for each protein. This summation of all correctly made decisions (regardless of whether or not it is a "yes" decision or a "no" decision) is divided by the number of decisions made. This gives a percentage that shows directly how changing entropy thresholds affects the predicting power of a given decision tree.

3.10 Decision Forest Parameter Setup

The ID3 decision tree, in this particular implementation and application, has three primary parameters, some of which have already been defined: label pivot, attribute range set, and entropy threshold[14]. The label pivot determines what range of labels, as described in section 3.6, are considered "yes" labels, and, alternatively, the range that denotes "no" labels. Naturally, the magnitude of this number has a large effect on the outcome of the decision trees. The attribute range set is composed of a short list of amino acid frequency ranges that serve as the classifying attributes. The length of this list and the distance between each of the bounds of the ranges also has a prominent effect on the decision tree, and its respective measures. The most sensitive parameter, however, is the entropy threshold, or, rather, the allowed level of randomness before the decision tree can make a decision. As one might expect, the closer the threshold is to 1.0, which is the maximum entropy a dataset can have, the shorter and less effective the decision tree becomes. Yet, an entropy threshold that is too restrictive (i.e. close to 0.0) would be detrimental to the purposes of this research for reasons explained more in depth in the Experimental Results section.

The parameters tested in this experiment include two label pivots (six and seven), two attribute range sets ({0-4, 5-7, 8-14, 15-29, 30-100}, {0-7, 8-14, 15- 29, 30-100}), and twenty-one different entropy thresholds, ranging from 0.0 to a maximum of 1.0 while incrementing by 0.05 units. All of these parameters were tested on all 799 protein clusters, such that 268,464 unique tuples were generated, giving various measures described in each of the sections above. The results of these tests are described in the Experimental Results section.

IV. Experimental Results

4.1 Parametric Criteria

For each of the 799 protein clusters generated by the FGK-Model, and for each of the parameter choices as described in section 3.10, an array of measures were recorded. This data was used for the purpose of deciding upon the most appropriate values for the three parameters for the decision tree implementation. These measures included ideal prediction accuracy, average node secondary structure similarity, average *yes node* secondary structural similarity, decision node count, yes decision node count, and number of proteins classified within those yes nodes. Also included was a range of values that counted the percentage of decision nodes that had a secondary structure similarity measure of over 90%, 90-80%, 80-70%, 70-60%, and less than 60% structural homology. These values were used to determine what combination of entropy threshold, attribute range set, and label pivot would produce the optimal output for this research, based on various criteria. Obviously, one vies for high ideal prediction accuracy, because it implies high actual prediction accuracy such that parametric combinations that yielded these were kept. Likewise, a secondary structural similarity measure that is greater is more desirable than one that is not, with more emphasis placed on those combinations that yielded high average *yes node* secondary structural similarity measures. This is because the nodes that belong to the cluster (i.e. "yes" nodes) are statistically more important.

Inverse to the other measures, it was decided that a *lower* node count (that is, the count of decisions made) would be more favorable. This is due to the fact that this research aims to find protein sequence motifs that transcend protein families. If the node count is too high, and approaches the number of proteins, this implies that each node represents approximately one protein. As each decision node, ideally, should represent a given motif among the proteins it represents, it makes no sense to have a system in which each node only represents one protein. This, in itself, implies higher entropy and fewer items in the attribute range list.

Finally, it was decided that those parameters that gave higher percentages of nodes that have 70% structural homology or above (see section 3.7), were ideal.

4.2 Parametric Results

Given the parameters, four distinct data sets were created from analyzing and averaging the appropriately weighted values from the 268,464 generated tuples. The graphs denoting these four data sets can be seen in the following figures. Ideal prediction accuracy, given by a red line refers to the measure described in section 3.9. Its value refers to the right y-axis. "Yes" node secondary structural similarity, given by a purple line, is exactly that, again referring to the right y-axis. Total secondary structural similarity, given by a green line, is the measure of all nodes' secondary structure. It, too, refers to the right y-axis. Total node count, a light blue line, is simply the number of all decision nodes, and it refers to the left y-axis. Since high quality nodes are important, we also show the percentage of nodes with greater than 90% structural similarity, given by a gray-blue line. This, again, is given by the right y-axis. Finally, the "yes" node count, denoted by an orange line, just refers to the number of yes decision nodes.

As one can see in each of the four figures, an entropy threshold of 0.75 is marked by a vertical red on the graph, noting the various measures at that entropy. One might note that the percentage of nodes with a 90% structural similarity line falls sharply on all four graphs *after* an entropy of 0.75. One might also note that ideal prediction accuracy follows a similar trend, but to a much less severe degree, just as average yes node secondary structure similarity measure. An entropy threshold of 0.75 also falls in the mid-range of the average node count, implying that it would not yield data too far dichotomized, nor would it yield completely random output. Keeping in mind all criteria spelled out in section 4.1, it would appear that an entropy threshold of 0.75 is, indeed, the most appropriate for this research.

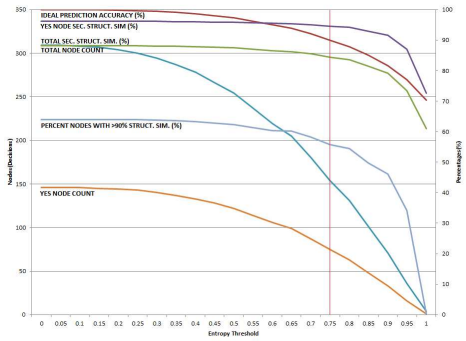

Figure 2 Seven Label Pivot, Large Attribute Range set

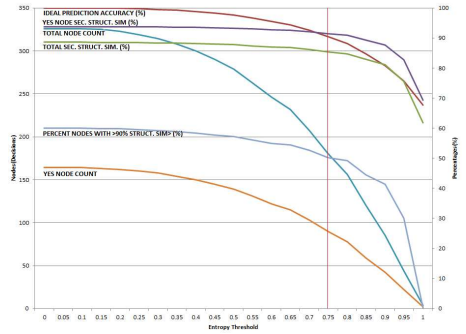

Figure 3 Six Label Pivot, Large Attribute Range Set

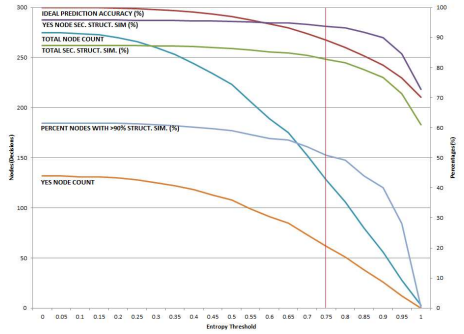

Figure 4 Seven Label Pivot, Reduced Attribute Range Set

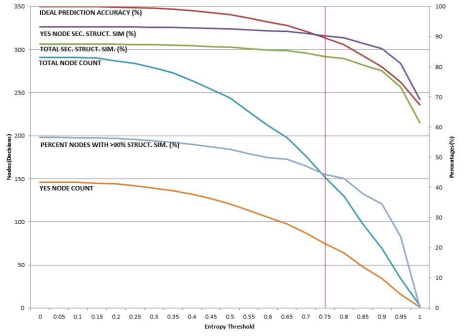

Figure 5 Six Label Pivot, Reduced Attribute Range Set

Parameters	>90%	90-80%	80-70%	70-60%	<60%
P7-R5	55.83%	15.56%	7.19%	9.18%	12.24%
P6-R5	50.27%	15.92%	13.05%	8.97%	11.80%
P7-R4	50.80%	17.38%	7.92%	10.00%	13.90%
P6-R4	44.39%	17.45%	14.57%	9.94%	13.65%

Table 1. Comparison of Decision Node Protein Secondary Structural Similarity Percentages.

To determine which label pivot and attribute range set is optimal, one can refer to the measures of nodal structural similarity percentages given in Table 1, which assumes our given entropy threshold of 0.75. In this table, P7 refers to a label pivot of seven, P6 refers to a label pivot of six, R5 refers to the large ({0-4, 5-7, 8-14, 15-29, 30-100}) attribute set, and alternatively, R4 refers to the small attribute range set. As one can see, taking only those percentages that refer to greater than 70% structural similarity (as, again, they can be considered structurally identical [15]), P6-R5 produces the best results, with P7-R4 producing the worst results. Note that while P6-R5 doesn't produce the optimal percentage of nodes with greater than 90% structural similarity, it does produce both the most over 70% *and* has the least percentage of nodes with less than 60% structural similarity. Taking in consideration other measures, such as node count and average yes node secondary structural similarity, P6-R5 consistently produces the most optimal output.

4.3 Example Decision Tree Result

Thus, given the parameters of a 0.75 entropy threshold, and the parametric combination denoted as P6-R5 (refer to section 4.2) one can produce a relatively simple decision tree to examine the effectiveness of the FGK-Decision Forest Model. The following figure examines a random file whose number of decisions was in the lower range, such that it could be easily displayed on paper. Note that this tree is not typical in that the average range for the node count with the given parameters is 150:

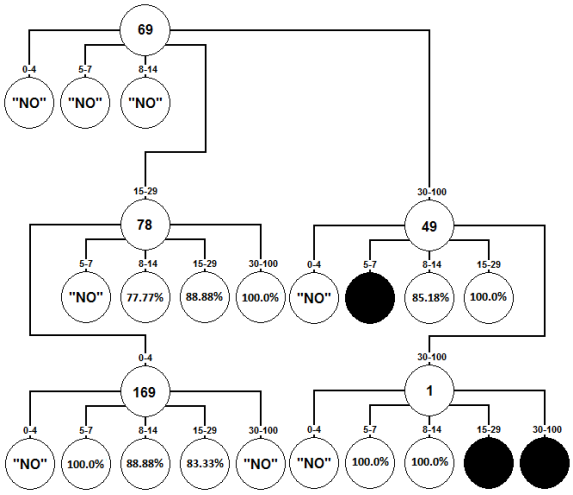

Figure 6 Granule 6-Cluster 93 Decision Tree

The method by which one would read Figure 6 is very simple. One starts at the top node, denoted here as '69,' and would work their way down to a given decision. The decision states whether or not a protein belongs to their FGK-Model generated clusters. The '69,' '169,' '78,' '1,' etc. are all dimensions for each protein generated by the sliding window technique. Each branch from each node denotes the attribute range that is used to further classify the data set. For instance, starting from the root node, '69,' if the frequency value of this dimension is between zero and four, then a "no" decision is made. In most cases, however, the decision tree must refer to

other dimensions and check their respective value before a decision can be made. The yes decision nodes are denoted as percentages, which detail the structural similarity of the proteins it describes. The black decision nodes denote a case in which no proteins in the training data could be represented by that particular path. These are interpreted as "no" decision nodes.

V. Conclusion

A newly proposed model, the FGK-Decision Forest Model, utilizes the data organizing prowess and robustness of granule computing granted by Fuzzy C-Means and Greedy K-Means clustering, and the clear and easily comprehendible analysis of the ID3 decision tree. Using this model, one splits the original protein data, generated by a sliding window technique, into various information granules of protein clusters via various iterations of Fuzzy C-Means and Greedy K-Means. Granted these clusters, a decision tree is generated for each. These decision trees each contain decisions that denote whether or not certain proteins belong to a given cluster. They also denote structural motifs, presented in the "yes" nodes of each tree. All of these decision trees come together to produce a "decision forest" in which one could potentially use to predict local tertiary structure by finding the decision tree and the motif contained therein that best fits the unknown protein, assuming paired tertiary structure data.

This paper focuses heavily on the parametric setup and analysis of the results of each. The three primary parameters tested were entropy, label pivot, and attribute range set. The entropy described the allowed randomness of the tree. It was set to 0.75, as it had the greatest tradeoff between all parametric criteria. The label was based on secondary structural similarity and the idea that 70% and greater secondary structural similarity was roughly identical. Two label pivots were tested, and a value of 6 was decided based on the quality analysis. The attribute range set was based on the frequency values produced by the sliding window technique. Two sets were tested, and the larger range set was used for its increased quality in regards to the parametric criteria.

A decision tree example is also shown, in which its usefulness for portraying clear and easily comprehendible analysis is examined. As each "yes" node denotes a structural motif, each "no" node denotes a set of proteins that need to be removed from the training data, and each black (that is, each node in which a decision was not generated) node denotes sections in which there are no structural motifs, it is clear that the decision tree is a promising method, at least graphically, for portraying protein data. Also, each decision tree can be used, without modification, to decide whether or not a protein belongs to the cluster represented by the protein, and with an associated prediction accuracy. This implies that the decision forest, as stated previously, can be used to generate local tertiary structural predictions with measurably accurate decisions.

While further development and research is needed to expand the flexibility and applicability of this model, it should be clear that it has potential to be adapted due to its promising robustness and efficiency, as well as the relative ease of comprehending its output, such that its analysis is not constrained to one field. With our proposed expansions on the original implementation, we believe this model will be used widely for the above mentioned reasons.

Acknowledgment

The work of Bernard Chen was supported in part by UCA's University Research Council (URC) and Summer Stipend. The work of D. Che was partially supported by President Research Fund at East Stroudsburg University of Pennsylvania.

References

[1] J.M. Chandonia, S.E. Brenner, "The impact of structural geonomics: expectations and outcomes," *Science*, vol. 311, pp. 347-351, 2006.

[2] A. L. Spek, "Structure validation in chemical crystallography," *Acta Crystallographica*, Section D, vol. 60, no. 4, pp. 148-155, 2004.

[3] G. Karp, *Cell and Molecular Biology: Concepts and Experiments*, 6th ed., New York: John Wiley & Sons Inc, 2009, pp. 52-66.

[4] N. Hulo, C.J.A.V. Sigrist, L. Saux, P.S. Langendijk-Genevaux, L. Bordoli, A. Gattiker, E. de Castro, P. Bucher, A. Bairoch, "Recent improvements to the PROSITE database," *Nucleic Acids Res.*, vol. 32, 2004.

[5] T.K. Attwood, M. Blythe, D.R. Flower, A. Gaulton, J.E. Mabey, N. Naudling, L. McGregor, A. Mitchell, G.M. Paine, R. Scordis, "PRINTS and PRINTS-S shed light on protein ancestry," *Nucleic Acid Res.,* vol. 30, no. 1, pp. 239-241, 2002.

[6] S. Henikoff, J.G. Henikoff, S. Pietrokovski, "Blocks+: a non redundant database of protein alignment blocks derived from multiple compilation," *Bioinformatics*, vol. 15, no. 6, pp. 417-479, 1999.

[7] O. Carugo, "Rapid Methods for Comparing Protein Structures and Scanning Structure Database," *Current Bioinformatics*, vol. 1, pp. 75-83, 2006.

[8] K.F. Han, D. Baker, "Global properties of the mapping between local amino acid sequence and local structure in proteins," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 93, no. 12, pp. 5814-5818, 1996.

[9] K.F. Han, D. Baker, "Recurring local sequence motifs in proteins," *Journal of Molecular Biology*," vol. 251, no. 1, pp. 176-187, 1995.

[10] C. Bystroff, V. Thorsson, D. Baker, "HMMSTR: a hidden Markov model for local sequence-structure correlations in proteins," *Journal of Molecular Biology*, vol. 301, pp. 173-190, 2000.

[11] B. Chen, P.C. Tai, R. Harrison, Y. Pan, "FGK model: An Efficient Granular Computing Model for Protein Sequence Motifs Information Discovery," *Proceedings of IASTED CASB Dallas,* pp. 56-61, 2006.

[12] B. Chen, M. Johnson, "Protein Local 3D Structure Prediction by Super Granule Support Vector Machines (Super GSVM)," *Proceedings of BMC Bioinformatics*, vol. 10, 2009.

[13] S. R. Safavian, D. Landgrebe, "A Survey of Decision Tree Classifier Methodology," *IEEE Trans.Systems, Man and Cybernetics*, vol. 21, no. 3, pp. 660-674, 1991.

[14] J.R. Quinlan, "Induction of Decision Trees," *Machine Learning*, vol. 1, pp. 81-106, 1986.

[15] W. Zhong, G. Altun, R. Harrison, P. C. Tai, Yi. Pan, "Improve K-Means Clustering algorithm for Exploring Local Protein Sequence motifs Representing Common Structural Property," *IEEE transactions on Nanobioscience*, vol. 14, no. 3, pp. 255-265, 2005.

[16] T.Y. Lin, "Data Mining and Machine Oriented Modeling: A Granular Computing Approach," *Journal of Applied Intelligence*, vol. 13, no. 2, pp. 113-124, 2002.

[17] Y.Y. Yao, "On Modeling data mining with granular computing," *Proceedings of COMPSAC 2001*, pp. 638-643, 2001.

[18] G. Wang, R. L. Dunbrack, Jr., "PISCES: a protein sequence culling server," *Bioinformatics*, vol. 19, no. 12, pp. 1589-1591, 2003.

[19] C. Sander, R. Schneider, "Database of similarity derived protein structures and the structure meaning of sequence alignment," *Proteins: Struct. Flunct. Genet.*, vol. 9, no. 1, pp. 56-68, 1991.

[20] W. Kabsch, C. Sander, "Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features," *Biopolymers*, vol. 22, pp. 2577-2637, 1983.