

FIK Model: Novel Efficient Granular Computing Model for Protein Sequence Motifs and Structure Information Discovery

Bernard Chen, Phang C. Tai, Robert Harrison and Yi Pan, *Senior Member, IEEE*

Abstract— Protein sequence motifs information is very important to the analysis of biologically significant regions. The conserved regions have the potential to determine the conformation, function and activities of the proteins. The main purpose of this paper is trying to obtain protein sequence motifs which are universally conserved and across protein family boundaries. Therefore, unlike most popular motif discovering algorithms, our input dataset is extremely large. As a result, an efficient technique is demanded. In this article, short recurring segments of proteins are explored by utilizing a novel granular computing strategy. First, Fuzzy C-Means clustering algorithm (FCM) is used to separate the whole dataset into several smaller informational granules and then succeeded by improved K-means clustering algorithm on each granule to obtain the final results. The structural similarity of the clusters discovered by our approach is studied to analyze how the recurring patterns correlate with its structure. Also, some biochemical references are included in our evaluation. To the best of our knowledge, it is the first time that the granular computing concept as well as the DBI measure for evaluation is introduced to this dataset. Compare with the latest research results, our method requires only twenty percent of the execution time and obtains even higher quality information of protein sequence motifs. The efficient and satisfactory results in our experiment suggests that our granular computing model which combined FCM and improved K-means may have a high chance to be applied in some other bioinformatics research fields and yield stunning results.

Index Terms—FIK Model, Fuzzy C-Means Clustering, Improved K-means clustering, Sequence Motif.

I. INTRODUCTION

Proteins can be regarded as one of the most important elements in the process of life; they could be separated into different families according to the sequential or structural similarities. The close relationship between protein sequence and structural plays an important role in current analysis and

prediction technologies. Therefore, understanding the hidden knowledge between protein structures and their sequences is an important task in today's bioinformatics research. The biological term sequence motif denotes a relatively small number of functionally or structurally conserved sequence patterns that occurs repeatedly in a group of related proteins. These motif patterns may be able to predict other protein's structural or functional area, such as enzyme-binding sites, DNA or RNA binding sites, prosthetic attachment sites ...etc.

PROSITE [8], PRINTS [9], and BLOCKS [10] are three most popular motifs databases. PROSITE sequence patterns are created from observation of short conserved sequences. Analysis of 3-D structures of PROSITE patterns suggests that recurrent sequence motifs imply common structure and function. Fingerprints from PRINTS contain several motifs from different regions of multiple sequence alignments, increasing the discriminating power to predict the existence of similar motifs because of individual parts of the fingerprint is mutually conditional [9]. Since sequence motifs from PROSITE, PRINTS, and BLOCKS are developed from multiple alignments, these sequence motifs only search conserved elements of sequence alignment from the same protein family and carry little information about conserved sequence regions, which transcend protein families [1].

Some of the commonly used tools for protein sequence motif discovering include MEME [11], Gibbs Sampling [12], and Block Maker [13]. Some newer algorithms include MITRA [14], ProfileBranching [15], and generic motif discovery algorithm for sequential data [16]. Users are asked to give several protein sequences, which may be required in FASTA format, as the input data while using these tools. Again, sequence motifs discovered by above methods may carry little information that crosses family boundaries, because the size of input dataset is limited.

Han and Baker have used the K-means clustering algorithm to find recurring protein sequence motifs [2]. They choose a set of initial points as the centroids by random [2]. Wei et al proposed an improved K-means clustering algorithm to obtain initial centroids location more wisely [1]. Due to the fact that the performance of K-means clustering is very sensitive to initial points selection, the results published by Wei et al has been improved in their experiment. The main reason that both above papers use K-means instead of some other more advanced clustering technology is due to the extremely large

Manuscript received December 20, 2006. This work was supported in part by the U.S. National Institutes of Health (NIH) under Grants R01 GM34766-17S1 and P20 GM065762-01A1.

Bernard Chen is with the Computer Science Department, Georgia State University, Atlanta, GA 30303-4110 USA (e-mail: bchen3@cs.gsu.edu)

P. C. Tai is with the Biology Department, Georgia State University, Atlanta, GA 30303-4110 USA (e-mail: biopct@langate.gsu.edu).

R. Harrison is with the Computer Science Department and the Biology Department, Georgia State University, Atlanta, GA 30303-4110 USA (e-mail: rharrison@cs.gsu.edu).

Y. Pan is with the Computer Science Department, Georgia State University, Atlanta, GA 30303-4110 USA (e-mail: pan@cs.gsu.edu).

input dataset. Since K-means is famous for its efficiency, other clustering methods with higher time and space costs may not be suitable for this task. To overcome the high computational cost caused by a huge input dataset, we proposed a granular computing model that utilized Fuzzy C-means clustering algorithm to divide the whole data space into several smaller subsets and then apply improved K-means algorithm to each subset to discover relevant information.

II. GRANULAR COMPUTING TECHNIQUES

A. Fuzzy C-Means Clustering Algorithm (FCM)

Fuzzy c-means (FCM) is a clustering algorithm which allows one segment of data to belong to one or more clusters. This method (developed by Dunn in 1973 [17] and improved by Bezdek in 1981 [18]) is frequently used in pattern recognition. The main purpose of this algorithm is trying to minimize the following objective function:

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|x_i - c_j\|^2, 1 \leq m < \infty$$

where m , the fuzzification factor, is any real number greater than 1, u_{ij} is the degree of membership of x_i in the cluster j , x_i is the i th of d -dimensional measured data, c_j is the d -dimension center of the cluster, and $\|\cdot\|$ is any norm expressing the similarity between any measured data and the center. Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership u_{ij} and the cluster centers c_j by:

$$u_{ij} = \frac{1}{\sum_{k=1}^C \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}$$

$$c_j = \frac{\sum_{i=1}^N u_{ij}^m \cdot x_i}{\sum_{i=1}^N u_{ij}^m}$$

This iteration will stop when $|U^{(k+1)} - U^{(k)}| < \epsilon$, where ϵ is a termination criterion, and k are the iteration steps. This procedure converges to a local minimum or a saddle point of J_m . [19]. The algorithm is described as following:

1. Initialize membership function matrix, $U^{(0)}$, and randomly select a set of initial centroids.
2. At k -step: update $U^{(k)}$ to $U^{(k+1)}$ by the function of u^{ij} .
3. Calculate the centroid information by c^j function.
4. Repeat step 2 until $|U^{(k+1)} - U^{(k)}| < \epsilon$ [19]

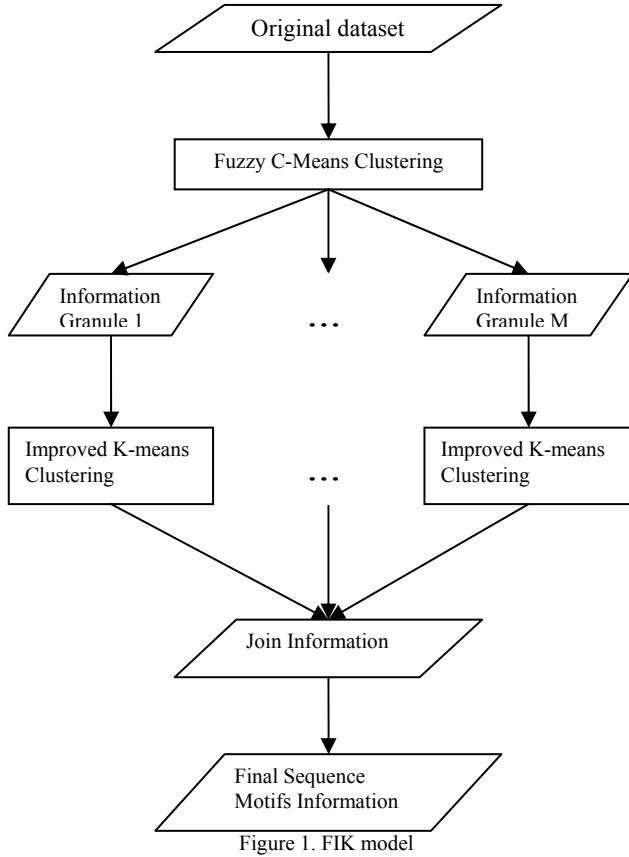
B. Improved K-Means Clustering Algorithm

This method is proposed by Wei et al [1] to overcome the potential problem of random initialization. It is a greedy initialization method that tries to choose suitable initial points so that final partitions can represent a more consistent and accurate result. In their experiment, the original random K-means clustering algorithm was performed five times. In each round, initial points which have the potential to form the cluster with high structural similarity are chosen for the improved K-means clustering algorithm. For each time a new potential initial center is chosen, its distance is checked against all points that are already selected in the initialization array. If the minimum distance of a new point is greater than the threshold distance, this point will be included in the initialization array; otherwise, this point is discarded and another potential initial centroid is tried until the desired number of centroids is chosen.

C. Novel Granular Computing Model

Granular computing represents information in the form of aggregates, also called “information granules” [20] [21]. For a huge and complicated problem, it uses the divide-and-conquer concept to split the original task into several smaller subtasks to save time and space complexity. Also, in the process of splitting the original task, it comprehends the problem without including all meaningless information. As opposed to traditional data-oriented numeric computing, granular computing is knowledge-oriented [21].

A granular computing based learning model called “Fuzzy-Improved-K-means model” (FIK model) is proposed in this work. This model works by building a set of information granules by FCM and then applying improved K-means clustering algorithm to obtain the final information. The improved K-means clustering algorithm has been slightly changed in this model. In order to obtain a more global result, instead of picking up initial seeds in each round of original K-means, we collect all five K-means results and then select the initial centroids, which not only has the potential to form the highly structural similarity clusters (>60%) but also recurrently appear for at least three times. While selecting those potential initial centroids, as long as they meet the criteria, we do not check the distance with other initial seeds. However, due to the recurrently appearing centroids limitation, we may not select all initial points by this method. Therefore, we usually obtained around one third starting centers that the informational granule required and get the other initial centroids randomly with distance check. Results with different distance thresholds are given in section four. Major advantages of FIK model are reduced time- and space-complexity, filtered outliers, and higher quality granular information results. We will present comparative results in section 4 of this paper. Figure 1 shows the sketch of the model.



At the first stage, all of data segment are clustered by Fuzzy C-Means into several “functional granules” by a certain membership threshold cut. In each functional granule, an improved K-means clustering is performed. At the final stage, we join the information generated by all granules and obtain the final sequence motifs information.

III. EXPERIMENT SETUP

A. Dataset

The dataset used in this work includes 2710 protein sequences obtained from Protein Sequence Culling Server (PISCES) [5]. No sequence in this database share more than 25% sequence identities. Sliding windows with 9 successive residues are generated from protein sequence. Each window represents one sequence segment of nine continuous positions. More than 560,000 segments are generated by this method. The frequency profile from the HSSP [3] is constructed based on the alignment of each protein sequence from the protein data bank (PDB) where all the sequences are considered homologous in the sequence database. We also obtained secondary structure from DSSP [4], which is a database of secondary structure assignments for all protein entries in the Protein Data Bank.

B. Representation of Sequence Segment

The sliding windows with nine successive residues are

generated from protein sequences. Each window corresponds to a sequence segment, which is represented by a 9×20 matrix plus additional nine corresponding secondary structure information obtained from DSSP. Twenty rows represent 20 amino acids and 9 columns represent each position of the sliding window. For the frequency profiles (HSSP) representation for 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. DSSP originally assigns the secondary structure to eight different classes. In this paper, we convert those eight classes into three classes based on the following method: H, G and I to H (Helices); B and E to E (Sheets); all others to C (Coils).

C. Distance Measure

According to Han et al [2] and other related researches [1], city block metric is more suitable for this field of study since it will consider every position of the frequency profile equally.

D. Structure Similarity Measure

Cluster’s average structure is calculated using the following

$$\text{formula: } \frac{\sum_{i=1}^{ws} \max(P_{i,H}, P_{i,E}, P_{i,C})}{ws}$$

Where ws is the window size and $P_{i,H}$ shows the frequency of occurrence of helix among the segments for the cluster in position i . $P_{i,E}$ and $P_{i,C}$ are defined in a similar way. If the structural homology for a cluster exceeds 70%, the cluster can be considered structurally identical [7]. If the structural homology for the cluster exceeds 60% and bellows 70%, the cluster can be considered weakly structurally homologous [1].

E. Davis-Bouldin Index (DBI) Measure

Besides using secondary structure information as a biological evaluation criterion, we also introduce an evaluation method used in computer science in this dataset for the first time. The DBI measure [6] is a function of the inter-cluster and intra-cluster distances. A good cluster result should reflect a relatively large inter-cluster distance and a relatively small intra-cluster distance. The DBI measure combines these two distance information into one function, which is defined as follows:

$$DBI = \frac{1}{k} \sum_{p=1}^k \text{MAX}_{p \neq q} \left\{ \frac{d_{int\ ra}(C_p) + d_{int\ ra}(C_q)}{d_{int\ er}(C_p, C_q)} \right\}, \text{ where}$$

$$d_{int\ ra}(C_p) = \frac{\sum_{i=1}^{n_p} \|g_i - g_{pc}\|}{n_p} \quad \text{and} \quad d_{int\ er}(C_p, C_q) = \|g_{pc} - g_{qc}\|$$

k is the total number of clusters, d_{intra} and d_{inter} denote the intra- cluster and inter-cluster distances respectively. n_p is the number of members in the cluster C_p . The intra-cluster distance defined as the average of all pair wise distance between the members in cluster P and cluster P's centroid, g_{pc} . The inter-cluster distance of two clusters is computed by the distance between two clusters' centroids. The lower DBI value indicates the higher quality of the cluster result.

F. Parameters Setup

In our experiment, 800 clusters are discovered. For the Fuzzy C-means clustering, the fuzzification factor is set to 1.05 and the number of clusters is equal to ten. This setting yielded the best results in our specific dataset. For example, if we set fuzzification factor as the same as above and cluster the whole dataset into 20 groups, the membership function cannot perform well and cause the result of each segment to have almost the same membership to every cluster. And if we further decrease the fuzzification factor, overflow may occur. In order to separate information granules from FCM results, the membership threshold is set to 12%. Using this value, we filter out around 15% of the dataset and assign the rest of the data to one or more clusters. The function that decides how many numbers of clusters should be 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 information granule k. n_k is the number of members belonging to information granule k. m is the number of clusters in FCM.

	Number of Members	Number of Clusters	Data Size
Granule 0	136112	151	99.9MB
Granule 1	68792	76	50.5MB
Granule 2	86094	95	63.2MB
Granule 3	65361	72	47.9MB
Granule 4	63159	70	46.3MB
Granule 5	120130	133	88.2MB
Granule 6	128874	142	94.6MB
Granule 7	4583	5	3.3MB
Granule 8	43254	47	31.7MB
Granule 9	5031	6	3.7MB
Total	721390	799	529MB
Original dataset	562745	800	413MB

Table 1 summary of results obtained by FCM

Table 1 is the summary of the results from FCM. Although the total data size increased from 413MB to 529MB and the total number of members increased from 562745 to 721390, we only deal with one information granule at a time. Therefore, we achieved the goal of reduced space-complexity.

G. Hardware and Software Information

All codes in this paper are written in Python, which is a dynamic object-oriented programming language, and executed under Microsoft Windows XP Command Prompt (simulated DOS environment). All time comparison results came from the same machine, which contains Pentium 4 CPU 1.7GHz with 1GB memory.

IV. EXPERIMENTAL RESULTS

A. Comparison of Execution Time

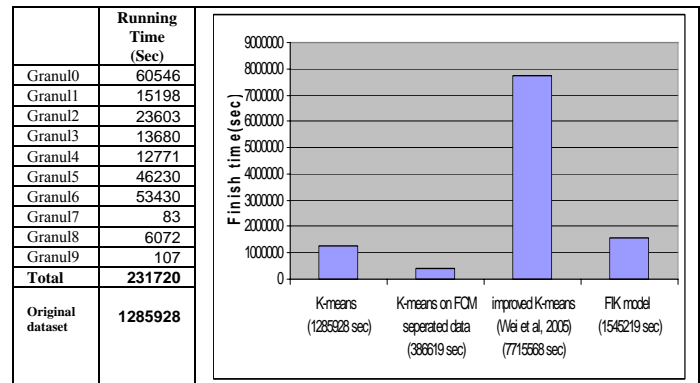


Table 2 Execution time comparison table

In table 2, the average K-means execution time for all information granules and the original dataset is given on the left column. On the right column, a graph that compares the average execution time for all methods mentioned in this paper is shown. From the graph, “K-means” represents the average execution time for applying original K-means algorithm on the intact dataset once. “K-means on FCM separated data” gives the average run time for executing original K-means algorithm on the information granules obtained by Fuzzy C-means clustering. The total execution time shown for “K-means on FCM separated data” on the graph equals to the sum of the execution time of all informational granules plus the time required by Fuzzy C-means clustering algorithm (154899 seconds). The third method, “Improved K-means” created by Wei et al in 2005, requires the original K-means to be executed five times and the sixth iteration to obtain the result. Without discussing the trivial details, basically, their method requires six iterations of K-means clustering algorithm on the original dataset. Therefore, the value shown on the graph equals to the “K-means” value times six. The last method, “FIK model”, is the model presented in this paper. The method to compute the total required time is similar to Wei’s method: the sum of

execution time on all information granules times six plus one iteration time required by FCM.

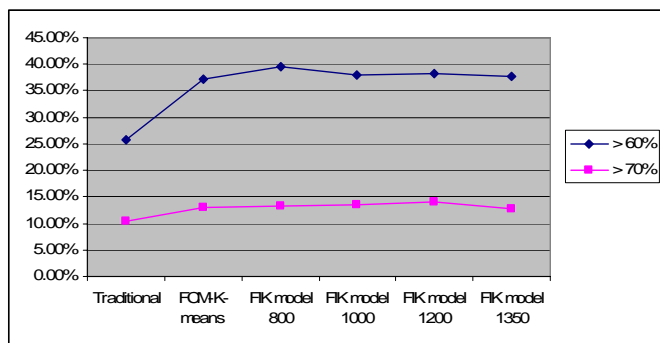
By comparing the execution times, our model requires only twenty percent of Wei's approach and almost equals to the time needed by original K-means clustering on whole dataset for one round. This result shows that the granular computing model really decreases the time-complexity of this task.

B. Sequence Motifs Quality Comparison

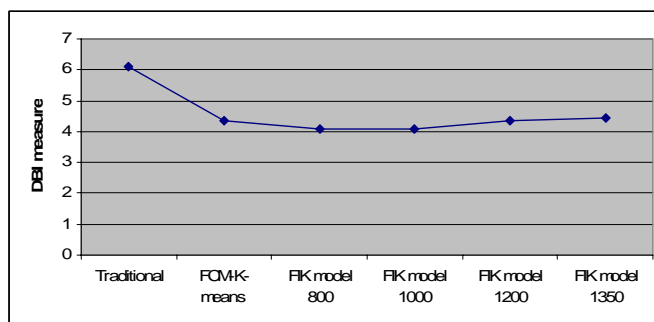
In table 3, the DBI measure and average percentage of sequence segments belonging to clusters with high structural similarity for different methods is given. The first column shows the different methods with different parameters. "Traditional" refers to the original K-means algorithm applied to the whole dataset. "FCM-K-means" indicates the original K-means clustering method applied to information granules generated by FCM. "FIK model 800" shows that the dataset is computed using the FIK model resulting in a distance of at least 800 between the initial centroids generated by improved tactics and the other centroids generated randomly. "FIK model 1000", "FIK model 1200" and "FIK model 1350" are defined similarly. The second column of Table 3 gives the average percentage of sequence segments belonging to the cluster with structural similarity greater than 60%. Similarly, the third column contains the average percentage of the clusters with the structural similarity higher than 70%. The last column shows the average DBI measure (The lower DBI value indicates the higher quality of the cluster result). Graph 1 and graph 2 are interpreted from table 3.

Different Methods	> 60%	> 70%	DBI Measure
Traditional	25.82%	10.44%	6.098
FCM-K-means	37.14%	12.99%	4.359
FIK model 800	39.42%	13.27%	4.095
FIK model 1000	38.05%	13.64%	4.090
FIK model 1200	38.29%	14.14%	4.331
FIK model 1350	37.67%	12.77%	4.460

Table 3 Comparison of DBI measure and percentage of sequence segments belonging to clusters with high structural similarity.



Graph 1 Comparison of percentage of sequence segments belonging to cluster with high structure similarity.



Graph 2 Comparison of DBI measure.

The results of Table 3 and both graphs 1 and 2 reveal that the quality of clusters improved dramatically by applying granular computing techniques which utilizes FCM to separate whole dataset into several information granules. The average percentage of clusters with structural similarity greater than 60% are increased more than ten percent, which translates to more than 80 meaningful sequence motifs that cannot be disclosed by traditional method but are discovered by our approach. The DBI measure also successfully decreased from 6 to 4.5-4.0, implying that our model not only generates more biologically meaningful results but that these results are supported by statistical/computer-science techniques. The improved K-means clustering algorithm also plays an important role in the FIK model. Although it increases only one to two percent of the structural similarity, the improved K-means algorithm can capture a more global result than the original one. Since the centroids that formed recurring high quality clusters may not always be chosen at random and could cause the information to be lost. The percentage of clusters with structural similarity higher than 70% is slightly improved with increasing the minimum distance threshold in improved K-means algorithm. However, once the distance threshold is set too high, the clustering result may suffer from both wasting too much time in choosing centroids and obtaining a degraded result. "FIK model 1350" is a good example of it. Since we have a larger dataset and different window sizes from Wei et al [1], we cannot directly compare the results. Nevertheless, we ameliorated the structural similarity of clusters more than 10%, while their best work increased only from 30.35% to 34.86%; our achievement is more dominant.

C. Sequence Motifs

The table 4 to 11 illustrates eight different sequence motifs generated by our method. Due to space limitation, we only present part of our recurring patterns information in this paper. However, all of clusters with 60% secondary structural similarity results generated by FIK-800 are available under <http://www.cs.gsu.edu/~cscbecx/Bioinformatics%20Information.htm>. The following format is used for representation of each motif table.

- The first row represents number of members belonging to this motif and the secondary

structural similarity.

- The first column stands for the position of amino acid profiles in each motif with window size nine.
- The second column expresses the type of amino acid frequently appeared in the given position. If the amino acids appearing with the frequency higher than 10%, they are indicated by upper case; If the amino acids appearing with the frequency between 8% and 10%, they are indicated by lower case.
- The third column corresponds to the hydrophobicity value, which is the summation of the frequencies of occurrence of Leu, Pro, Met, Trp, Ala, Val, Phe, and Ile.
- The last column indicates the representative secondary structure to the position.

Number of segments: 1581 Structure homology: 80.87%			
No	Noticeable Amino Acid	H	S
1	A K E d	0.24	H
2	A K E	0.35	H
3	A	0.86	H
4	A R K l e	0.42	H
5	A K E r d	0.23	H
6	I A v l	0.54	H
7	V L I a	0.76	H
8	R K E a	0.26	H
9	A K E	0.28	H

Table 4 Helices motif with conserved A

Number of segments: 1169 Structure homology: 79.76%			
No	Noticeable Amino Acid	H	S
1	a K E d	0.23	H
2	a K E d	0.23	H
3	K E	0.25	H
4	a R K q E	0.26	H
5	A K q E	0.24	H
6	a R K E	0.26	H
7	v L k e	0.52	H
8	l a K E	0.33	H
9	a r K E	0.24	H

Table 5 Helices motif with conserved K and E

Number of segments: 945 Structure homology: 67.01%			
No	Noticeable Amino Acid	H	S
1	a r K E	0.33	H
2	a R K E d	0.23	H
3	L A	0.72	H
4	V L I	0.66	H
5	A r K E d	0.25	H
6	A r K E	0.27	C
7	G	0.05	C
8	V L I a	0.61	C
9	V L I	0.53	C

Table 6 Helices-Coil motif

Number of segments: 1271 Structure homology: 62.91%			
No	Noticeable Amino Acid	H	S
1	G A S t	0.36	C
2	g A S t	0.37	C
3	G A S t	0.38	C
4	G A S t	0.31	C
5	G A S t	0.35	C
6	G A S t	0.32	C
7	G A s	0.33	C
8	G A S t	0.34	C
9	G A s t	0.37	C

Table 7 Coli motif with conserved G, A, S and T

Number of segments: 903 Structure homology: 71.81%			
No	Noticeable Amino Acid	H	S
1	V L I	0.78	E
2	V	0.47	E
3	V I	0.91	E
4	V I I	0.49	E
5	a s t	0.36	E
6	a s n D	0.25	C
7	G s E n D	0.21	C
8	G s E D	0.25	C
9	g E	0.32	C

Table 8 Sheet-coil motif

Number of segments: 598 Structure homology: 65.44%			
No	Noticeable Amino Acid	H	S
1	a s e	0.36	C
2	G N D	0.37	C
3	G a	0.38	C
4	R K E d	0.31	C
5	V L I	0.35	E
6	V L I	0.32	E
7	a k E	0.33	E
8	V L I	0.34	E
9	p S t D	0.37	E

Table 9 Coil-sheet motif

Number of segments: 645 Structure homology: 70.80%			
No	Noticeable Amino Acid	H	S
1	a R K E	0.22	H
2	A R K E	0.23	H
3	L y A	0.41	C
4	G n	0.10	C
5	V L I A	0.66	C
6	p t r K E d	0.21	C
7	V l a	0.56	E
8	V L I	0.83	E
9	V L I	0.68	E

Table 10 Helices-coil-sheet motif

Number of segments: 659 Structure homology: 69.09%			
No	Noticeable Amino Acid	H	S
1	A S T	0.46	H
2	V A s t	0.53	H
3	G A S	0.55	H
4	v l A S t	0.52	H
5	V l i A s	0.56	H
6	v l A	0.56	H
7	A	0.53	H
8	V L I a	0.79	H
9	V L i A	0.62	H

Table 11 Hydrophobic Helices motif

V. FUTURE WORKS

Aside from the issues that we discussed above, there are several interesting future works that remain to be explored. In this study, the cluster number of 800 may not be optimal. This problem is also addressed by Wei et al. Compared with their research, finding optimal cluster number for each information granule is much easier than the whole data space. With the information about the optimal cluster number, clustering results maybe potentially closest to underlying distribution patterns of the sample space [1]. Sequence motifs with different window sizes are also popular problems to engage in. There are several motifs extraction algorithms proposed in this field. But most of their data spaces are much smaller than ours. How to combine these methods efficiently is also a research topic. Last but not least, applying some advanced clustering algorithm with high computational cost on each information granule to further improve our FIK model is also an important future direction.

VI. CONCLUSION

In this study, a novel granular computing model which combines Fuzzy C-means and Improved K-means clustering algorithm has been proposed to solve high computational cost problems. In this model, we utilize fuzzy clustering to split the whole dataset into several information granules and analyze each granule by K-means clustering algorithm with a more advanced method of initializing centroids. Analysis of sequence motifs also shows that the granular computing technology may detect some subtle sequence information overlooked by K-means clustering algorithm alone. It is the first time that our research introduced granular computing concept into this biological meaningful dataset. Also, we showed that DBI measure is suitable as an evaluation method. Since our FIK model is capable of decreasing time and space complexity, filtering outliers, and capturing better results, we believe this novel strategy is a very powerful tool for bioinformatics research involving an extremely large database.

VII. ACKNOWLEDGEMENT

The authors would like to thank Dr. Wei Zhong et al for

sharing information and helping this work. This research was supported in part by the U.S. National Institutes of Health (NIH) under grants R01 GM34766-17S1 and P20 GM065762-01A1, and the U.S. National Science Foundation (NSF) under grants CCF-0514750 and ECS-0334813. This work was also supported by the Georgia Cancer Coalition and used computer hardware supplied by the Georgia Research Alliance.

REFERENCES

- [1] W. Zhong, G. Altun, R. Harrison, P. C. Tai and Yi. Pan, "Improved K-Means Clustering algorithm for Exploring Local Protein Sequence motifs Representing Common Structural Property", IEEE transactions on Nanobioscience, vol4, no.3, pp. 255-265. 2005
- [2] Karen F. Han and David Baker, "Recurring Local Sequence Motifs in Proteins," J. Mol. Biol, vol. 251 pp. 176-187
- [3] C. Sander and R. Schneider, "Database of similarity derived protein structures and the structure meaning of sequence alignment," *Proteins:Struct. Funct. Genet.*, vol.9 no. 1, pp. 56-68, 1991.
- [4] W. Kabsch and C. Sander, "Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features," *Biopolymers*, vol. 22, pp. 2577-2637, 1983.
- [5] G. Wang and R. L. Dunbrack, Jr., "PISCES: a protein sequence-culling server," *Bioinformatics*, vol. 19, no. 12, pp.1589-1591,2003
- [6] Davies, D.L. and Bouldin, D.W., "A cluster separation measure.," IEEE Trans. Pattern Recogn. Machine Intell., 1, 224-227, 1979.
- [7] C. Sander and R. Schneider, "Database of similarity derived protein structures and the structural meaning of sequence alignment," *Proteins: Struct. Funct. Genet.*, vol. 9, no.1, pp. 56-68, 1991
- [8] N. Hulo, C. J. A. Sigrist, V. Le Saux, P. S. Langendijk-Genevaux, L. Bordoli, A. Gattiker, E. De Castro, P. Bucher, and A. Bairoch, "Recent improvements to the PROSITE database," *Nucleic Acids Res.*, vol. 32, Database issue: D134-137, 2004
- [9] T. K. Attwood, M. Blythe, D. R. Flower, A. Gaulton, J. E. Mabey, N. Naudling, L. McGregor, A. Mitchell, G. Moulton, K. Paine, and P. Scordis, "PRINTS and PRINTS-S shed light on protein ancestry," *Nucleic Acid Res.* vol. 30, no. 1, pp. 239-241, 2002
- [10] S. Henikoff, J. G. Henikoff and S. Pietrokovski, "Blocks+: a non redundant database of protein Alignment blocks derived from multiple compilation," *Bioinformatics*, vol. 15, no. 6, pp. 417-479, 1999.
- [11] Bailey, T.L. and Elkan, C. Fitting a mixture model by expectation Maximization to discover motifs in biopolymers. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 2, 28-36. 1994.
- [12] Lawrence, C.E. et al. "Detecting subtle sequence signals: a Gibbs Sampling strategy for multiple alignment." *Science*, 262, 208-214. 1993
- [13] Henikoff, S. et al. "Automated construction and graphical presentation of Protein blocks from unaligned sequences." *Gene*, 163, GC17-GC26, 1995
- [14] Eskin, E. and Pevzner, P.A. Finding composite regulatory patterns in DNA sequences. *Bioinformatics*, 18 ((Suppl. 1)), 354-363, 2002
- [15] Price, A et al. "Finding subtle motifs by branching from sample strings." *Bioinformatics*, 19 (Suppl. 2), II149-II155, 2003
- [16] Kyle L. Jensen et al, "A Generic motif discovery algorithm for sequential data", *Bioinformatics*, vol 22, no.1, pp. 21-28, 2006.
- [17] J. C. Dunn: "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", *Journal of Cybernetics* 3: 32-57, 1973
- [18] J. C. Bezdek: "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, New York, 1981.
- [19] http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/cmeans.html
- [20] T. Y. Lin, "Data Mining and Machine Oriented Modeling: A Granular Computing Approach," *Journal of Applied Intelligence*, Kluwer, Vol 13, No 2, 113-124, 2002.
- [21] Y.Y. Yao, "On Modeling data mining with granular computing," *Proceedings of COMPSAC 2001*, pp.638-643, 2001.