7-17-2009

Discovery and Extraction of Protein Sequence Motif Information that Transcends Protein Family Boundaries

Bernard Chen

DISCOVERY AND EXTRACTION OF PROTEIN SEQUENCE MOTIF INFORMATION THAT TRANSCENDS PROTEIN FAMILY BOUNDARIES

by

BERNARD CHEN

Under the direction of Dr. Yi Pan

ABSTRACT

Protein sequence motifs are gathering more and more attention in the field of sequence analysis. The recurring patterns have the potential to determine the conformation, function and activities of the proteins. In our work, we obtained protein sequence motifs which are universally conserved across protein family boundaries. Therefore, unlike most popular motif discovering algorithms, our input dataset is extremely large. As a result, an efficient technique is essential. We use two granular computing models, Fuzzy Improved K-means (FIK) and Fuzzy Greedy K-means (FGK), in order to efficiently generate protein motif information. After that, we develop an efficient Super Granular SVM Feature Elimination model to further extract the motif information. During the motifs searching process, setting up a fixed window size in advance may simplify the computational complexity and increase the efficiency. However, due to the fixed size, our model may deliver a number of similar motifs simply shifted by some bases or including mismatches. We develop a new strategy named Positional Association Super-Rule to confront the problem of motifs generated from a fixed window size. It is a combination approach of the super-rule analysis and a novel

Positional Association Rule algorithm. We use the super-rule concept to construct a Super-Rule-Tree (SRT) by a modified HHK clustering, which requires no parameter setup to identify the similarities and dissimilarities between the motifs. The positional association rule is created and applied to search similar motifs that are shifted some residues. By analyzing the motifs results generated by our approaches, we realize that these motifs are not only significant in sequence area, but also in secondary structure similarity and biochemical properties.

INDEX WORDS: protein sequence motif, FIK model, FGK model, Super GSVM-FE, HHK clustering algorithm, Positional Association Rule, Super-Rule.

DISCOVERY AND EXTRACTION OF PROTEIN SEQUENCE MOTIF INFORMATION THAT TRANSCENDS PROTEIN FAMILY BOUNDARIES

by

BERNARD CHEN

A Dissertation Submitted in Partial Fulfillment of the Requirements of the Degree of

Doctor of Philosophy

in the College of Arts and Science

Georgia State University

2008

DISCOVERY AND EXTRACTION OF PROTEIN SEQUENCE MOTIF INFORMATION THAT TRANSCENDS PROTEIN FAMILY BOUNDARIES

by

BERNARD CHEN

Committee Chair: Yi Pan
Committee: Phang C. Tai
Robert. W. Harrison
Yanqing Zhang

Electronic Version Approved:

Office of Graduate Studies
College of Arts and Science
Georgia State University
May 2008

DEDICATION

To my mother 張秀月 and my late father 陳世明

ACKNOWLEDGEMENTS

It is a great honor for me to peruse my Ph.D. degree at Georgia State University. Fortunately, I have met many people during my graduate student life.

First of all, I would like to thank my advisor, Dr. Yi Pan, for all of his guidance, support, tolerance, and for his unending patience. Based on his supervision, I learned and enjoyed a lot in my life as a Ph.D. student; and more importantly, he has leaded me toward to a bright future. I have been very lucky to have him as my advisor.

I would like to thank Dr. Phang C. Tai. He is a key reason I came to study in the United States. His support means a tremendous amount to me. He is the perfect example of how to be a successful professor and a great person. I am very grateful that I have not let him down during the past years.

I would like to thank Dr. Robert Harrison and Dr. Yanqing Zhang for all of the help they have given. They have provided me with many new research ideas and have also corrected my mistakes. They can always see what I missed and give endless help when I need it. It is my great pleasure to work with them.

I would like to thank the Computer Science Department at the Georgia State University and the Molecular Basis of Disease (MBD) fellowship at GSU for their support during my graduate studies. I would specially like to thank Dr. Raj Sunderraman who has helped me and so many other students in our department in many different ways. He has offered several great classes for me to teach. Based on those teaching experiences, I have refined my teaching style and sharpen my public speaking skills. He is the best graduate coordinator.

Finally, I would like to thank my father, my mother and my family. They have been the greatest support to me all my life and I feel very lucky and happy to have such a great family.

I would also like to thank all my friends in both Taiwan and the United States. It is because of the people I have thanked that I have made it where I am today.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

1. SVM - Support Vector Machine
2. NMR - Nuclear Magnetic Resonance
3. BLAST - The Basic Local Alignment Search Tool
4. PISCES - Protein Sequence Culling Server
5. HSSP - Homology-derived Secondary Structure of Proteins
6. PDB - Protein Data Bank
7. DSSP - Dictionary of Secondary Structure of Proteins
8. BLOSUM - BLOcks of Amino Acid SUbstitution Matrix substitution matrix
9. FIK - Fuzzy Improved K-means
10. FGK - Fuzzy Greedy K-means
11. Super GSVM-FE - Super Granule Support Vector Machine Feature Elimination
12. SRT - Super-Rule-Tree
13. HHK - Hybrid Hierarchical K-means

CHAPTER 1

INTRODUCTION

Bioinformatics involves the use of several different techniques, including Computer Science, Data Mining, Computational Intelligence, Statistics, Applied Mathematics, Chemistry, and Biochemistry, to solve problems of Molecular Biology. The work of Bioinformatics emphasizes the creation and advancement of algorithms to extract useful information from biological data. Since the problems in this field usually start with biological issues and then move to computational domains, the major challenge of Bioinformatics research is that researchers are required to be familiar with more than two disciplines of knowledge. In this work, we follow the same trend: First, we explain the biological terminologies, declare the goal of our research, clarify the challenges we confront, and give details to our experimental setups. Then, we move to bioinformatics algorithms and techniques including Granular Computing, Fuzzy Logic, Clustering, Feature Elimination, Ranking SVM, Super-Rules, Association Rules and High Performance Computing.

1.1 The Central Dogma of the Molecular Biology

The Central Dogma of Molecular Biology[1] describes that each gene in the DNA contains all the required information for constructing proteins. Three different processes are responsible for the inheritance of genetic information and for the conservation from one form to another[2]:

- **Replication**: A double strand DNA duplicates itself to generate another identical replica so that the DNA → RNA → Protein cycle can repeat during the new generation of cell or organisms.
- **Transcription**: A double strand DNA segment is transferred into a single strand newly assembled messenger RNA (mRNA).
- **Translation**: Eventually, the mature mRNA is translated into a sequence of amino acids as the formation of Protein.

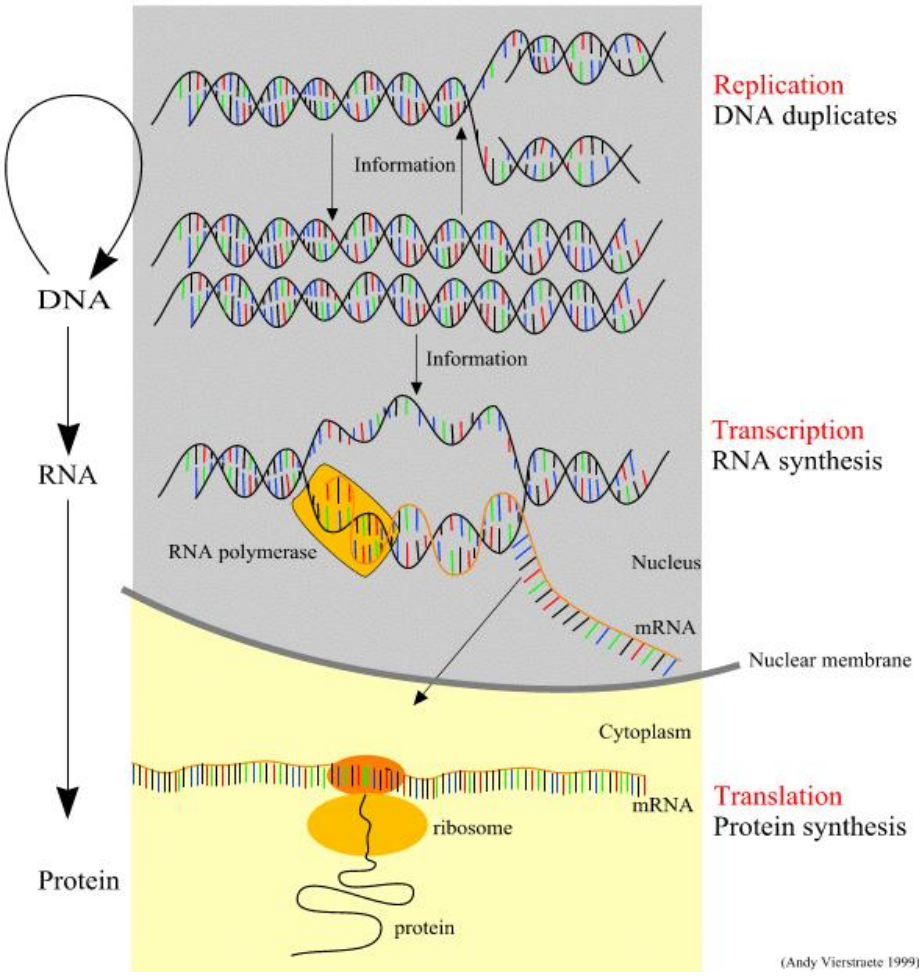


Figure 1.1 The illustration of the Central Dogma [2].

1.2 Levels of Structure in Proteins

Since the major biological problem we face in this work relates to the proteins, we need to explain four levels of structure in proteins: 20 different amino acids are the basic residues to construct proteins. The sequence of the amino acid is regarded as the **Primary Structure**. **Secondary Structure** is the arrangement in space of the atoms in the backbone of the polypeptide chain. The α-helix and β-sheet are two different types of secondary structure. **Tertiary Structure** includes the three-dimensional arrangement of all the atoms in the protein, including those in the side chains and in any prosthetic groups (groups of atoms other than amino acids) [3]. A protein can consist of multiple polypeptide chains called subunits. The arrangement of subunits with respect to one another is the **Quaternary Structure [3]**. Figure 1.2 gives a simple demonstration for all levels of the protein structure.

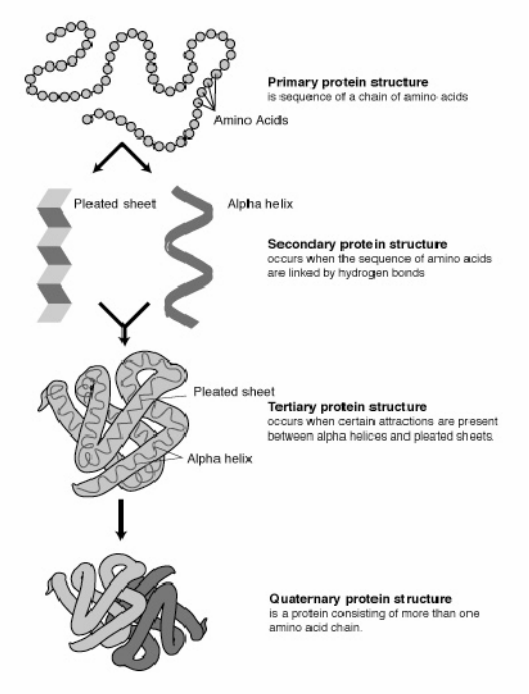


Figure 1.2 Protein Structure, from Primary Structure to Quaternary Structure [4]

1.3 Protein Sequence Motif

Proteins can be regarded as one of the most important elements in the process of life; they can be grouped into different families according to the sequential or structural similarities. Many biochemical tests suggest that a sequence determines conformation completely, because all the information that is necessary to specify protein interaction sites with other molecules is embedded into its amino acid sequence. The close relationship between protein sequence and structure plays an important role in current analysis and prediction technologies. Therefore, understanding the hidden relationships between protein structures and their sequences is an important task in modern bioinformatics research. The biological term sequence motif denotes a relatively small number of functionally or structurally conserved sequence patterns that occur repeatedly in a group of related proteins. These motif patterns may be able to predict the structural or functional area of other proteins, such as enzyme-binding sites, DNA or RNA binding sites, prosthetic attachment sites, protein-protein interaction sites etc.

1.4 Related Researche on Protein Sequence Motifs

PROSITE [5], PRINTS [6], and BLOCKS [7] are three of the most popular motif databases. PROSITE is a method of determining the function of uncharacterized proteins translated from genomic or cDNA sequences. It consists of a database of biologically significant sites and patterns formulated in such a way that with appropriate computational tools it can rapidly and reliably identify to which known family of protein (if any) the new sequence belongs [5]. Analysis of 3-D structures of PROSITE patterns suggests that recurrent sequence motifs imply common structure and function. Fingerprints, a group of conserved motifs used to characterize a

protein family, from PRINTS contain several motifs from different regions of multiple sequence alignments, increasing the discriminating power to predict the existence of similar motifs because individual parts of the fingerprint are mutually conditional [6]. The blocks are multiply aligned ungapped segments corresponding to the most highly conserved regions of proteins. The BLOCKS database is made automatically by looking for the most highly conserved regions in groups of proteins documented in the PROSITE Database [7]. 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 [8].

The commonly used tools for protein sequence motif discovering include MEME [9], Gibbs Sampling [10], and Block Maker [11]. Some newer algorithms include MITRA [12], ProfileBranching [13], and generic motif discovery algorithm for sequential data [14]. When using these tools, users are asked to give several protein sequences, normally presented in the FASTA format, as the input data. Again, sequence motifs discovered by the above methods may carry little information that crosses family boundaries, because the size of the input dataset is limited.

Some researchers have tried to obtain protein sequence motifs which are universally conserved across protein family boundaries. In order to achieve this goal, the input dataset has to be big enough to somehow represent all known protein sequences. Han and Baker [15] have used the K-means clustering algorithm to find recurring protein sequence motifs. They choose a set of initial points as the centroids at random. Zhong et al [8] proposed an improved K-means clustering algorithm to obtain initial centroid locations more wisely. Due to the fact that the performance of K-means clustering is very sensitive to initial point selection, the experiment of

Zhong et al [8] shows improved results. The main reason that the above authors used K-means clustering algorithm instead of some other more advanced clustering technology is because of the extremely large input dataset. Since K-means is known for its efficiency, other clustering methods with higher time and space costs may not be suitable for this task.

1.5 The Major Goal of Our Work

The main purpose of this work is to obtain and extract protein sequence motifs which are universally conserved and across protein family boundaries.

1.6 Challenges

There are four fundamental challenges when working to obtain biologically meaningful protein sequence motifs which are universally conserved across protein family boundaries:

- How to obtain the dataset and deal with this large volume of data.
- How to apply high performance computing techniques to speedup the searching time.
- How to deal with some noise data which maybe useless or even harmful.
- How to find the relations between motifs and motifs.

1.7 Organizations

In this first chapter, we introduce some basic biological information and assert our research goal, as well as difficulties we confront. The next chapter explains how we setup our experimental dataset and evaluation methods. Chapter 3 and chapter 4 focuses on efficiently

discovering high quality protein sequence motifs from the large dataset. Two granule computing models (Fuzzy Improved K-means and Fuzzy Greedy K-means model) have been proposed to search the protein recurring patterns. Chapter 5 focuses on extracting obtained motif information by a novel Super Granule Support Vector Machine model. The relation between motifs and motifs has been mined by the Super-Rule-Tree structure and the Positional Association Rule algorithm described in chapter 6 and 7, respectively. Chapter 8 expends on how high performance computing is involved in the protein sequence motifs discovering process. Finally, a summary of our achievements and future works are presented in the last chapter.

CHAPTER 2

EXPERIMENT SETUP FOR DISCOVERING PROTEIN SEQUENCE MOTIFS

2.1 Dataset

Since the major purpose of this work is to obtain protein sequence motif information across protein family boundaries, the dataset of our work is supposed to collect all known protein sequences. However, without a systematic approach, it is very difficult to extract useful knowledge from an extremely large volume of data. The dataset used in this work includes 2710 PDB protein sequences obtained from Protein Sequence Culling Server (PISCES) [16]. No sequence in this database shares more than 25% sequence identity. The frequency profile from the HSSP [17] 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. The basic principle we use us to choose representative protein files from the whole PDB database, and then use the profile in HSSP to expand each file. Figure 2.1 shows the picture of this idea. In the end of each HSSP file, it calculates the occurrence percentage of every amino acid on each sequence position. An example of an HSSP file is given in Figure 2.2. We also obtained secondary structure from DSSP [18], which is a database of secondary structure assignments for all protein entries in the Protein Data Bank, for evaluation purposes.

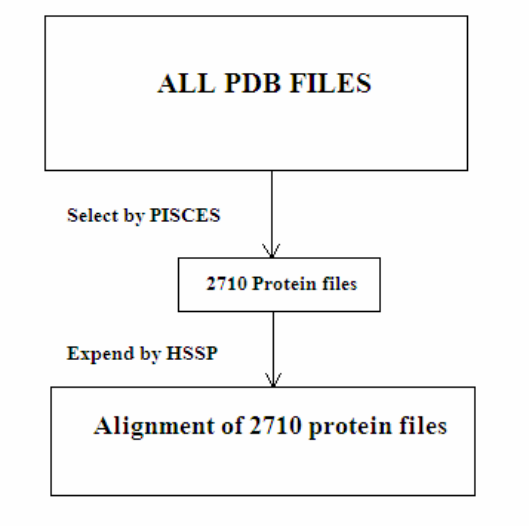


Figure 2.1 The approach that we use to select our experimental dataset

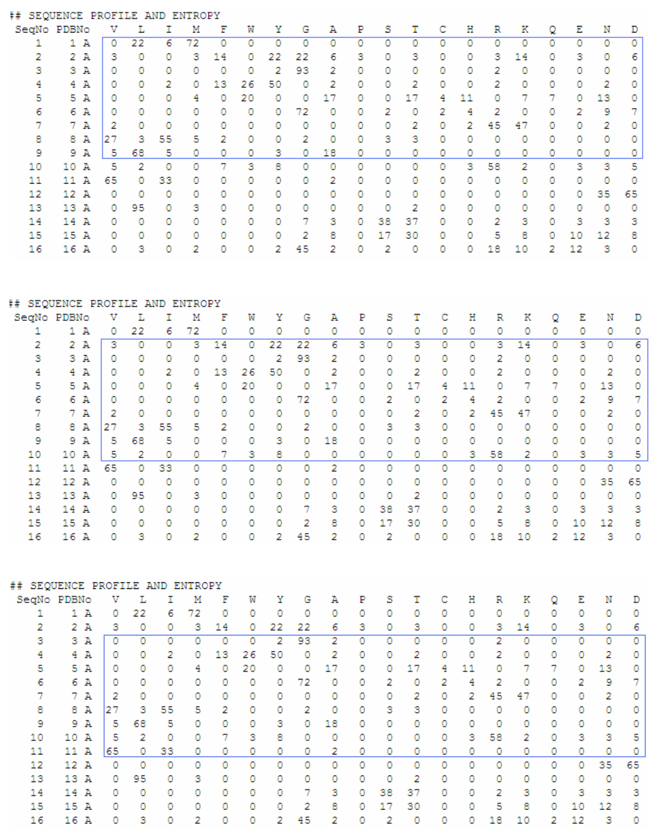
```
## SEQUENCE PROFILE AND ENTROPY
 SeqNo PDBNo    V    L    I    M    F    W    Y    G    A    P    S    T    C    H    R    K    Q    E    N    D
    1     1 A    0   22    6   72    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    2     2 A    3    0    0    3   14    0   22   22    6    3    0    3    0    0    3   14    0    3    0    6
    3     3 A    0    0    0    0    0    0    2   93    2    0    0    0    0    0    2    0    0    0    0    0
    4     4 A    0    0    2    0   13   26   50    0    2    0    0    2    0    0    2    0    0    0    2    0
    5     5 A    0    0    0    4    0   20    0    0   17    0    0   17    4   11    0    7    7    0   13    0
    6     6 A    0    0    0    0    0    0    0   72    0    0    2    0    2    4    2    0    0    2    9    7
    7     7 A    2    0    0    0    0    0    0    0    0    0    0    2    0    2   45   47    0    0    2    0
    8     8 A   27    3   55    5    2    0    0    2    0    0    3    3    0    0    0    0    0    0    0    0
    9     9 A    5   68    5    0    0    0    3    0   18    0    0    0    0    0    0    0    0    0    0    0
   10    10 A    5    2    0    0    7    3    8    0    0    0    0    0    0    3   58    2    0    3    3    5
   11    11 A   65    0   33    0    0    0    0    0    2    0    0    0    0    0    0    0    0    0    0    0
   12    12 A    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0   35   65
   13    13 A    0   95    0    3    0    0    0    0    0    0    0    2    0    0    0    0    0    0    0    0
   14    14 A    0    0    0    0    0    0    0    7    3    0   38   37    0    0    2    3    0    3    3    3
   15    15 A    0    0    0    0    0    0    0    2    8    0   17   30    0    0    5    8    0   10   12    8
   16    16 A    0    3    0    2    0    0    2   45    2    0    2    0    0    0   18   10    2   12    3    0
```

Figure 2.2 Part of 1b25 HSSP file. On the left hand side, it gives the PDB protein sequence number and chain number. For each amino acid location, it gives 20 numbers to represent the percentage of the occurrence of each amino acid.

2. 2 Representation on Data Segment

The sliding windows with nine successive residues are generated from protein sequences. Each window represents one sequence segment of nine continuous positions. More than 560,000 segments are generated by this method. Figure 2.3 shows how we apply the sliding window

technique on the HSSP file. 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 work, we convert those eight classes into three based on the following method: H, G and I to H (Helices); B and E to E (Sheets); all others to C (Coils).


SEQUENCE PROFILE AND ENTROPY

SeqNo	PDBNo		V	L	I	M	F	W	Y	G	A	P	S	T	C	H	R	K	Q	E	N	D
1	1	A	0	22	6	72	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	2	A	3	0	0	3	14	0	22	22	6	3	0	3	0	0	3	14	0	3	0	6
3	3	A	0	0	0	0	0	0	2	93	2	0	0	0	0	0	2	0	0	0	0	0
4	4	A	0	0	2	0	13	26	50	0	2	0	0	2	0	0	2	0	0	0	2	0
5	5	A	0	0	0	4	0	20	0	0	17	0	0	17	4	11	0	7	7	0	13	0
6	6	A	0	0	0	0	0	0	0	72	0	0	2	0	2	4	2	0	0	2	9	7
7	7	A	2	0	0	0	0	0	0	0	0	0	0	2	0	2	45	47	0	0	2	0
8	8	A	27	3	55	5	2	0	0	2	0	0	3	3	0	0	0	0	0	0	0	0
9	9	A	5	68	5	0	0	0	3	0	18	0	0	0	0	0	0	0	0	0	0	0
10	10	A	5	2	0	0	7	3	8	0	0	0	0	0	0	3	58	2	0	3	3	5
11	11	A	65	0	33	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
12	12	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	65
13	13	A	0	95	0	3	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
14	14	A	0	0	0	0	0	0	0	7	3	0	38	37	0	0	2	3	0	3	3	3
15	15	A	0	0	0	0	0	0	0	2	8	0	17	30	0	0	5	8	0	10	12	8
16	16	A	0	3	0	2	0	0	2	45	2	0	2	0	0	0	18	10	2	12	3	0

SEQUENCE PROFILE AND ENTROPY

SeqNo	PDBNo		V	L	I	M	F	W	Y	G	A	P	S	T	C	H	R	K	Q	E	N	D
1	1	A	0	22	6	72	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	2	A	3	0	0	3	14	0	22	22	6	3	0	3	0	0	3	14	0	3	0	6
3	3	A	0	0	0	0	0	0	2	93	2	0	0	0	0	0	2	0	0	0	0	0
4	4	A	0	0	2	0	13	26	50	0	2	0	0	2	0	0	2	0	0	0	2	0
5	5	A	0	0	0	4	0	20	0	0	17	0	0	17	4	11	0	7	7	0	13	0
6	6	A	0	0	0	0	0	0	0	72	0	0	2	0	2	4	2	0	0	2	9	7
7	7	A	2	0	0	0	0	0	0	0	0	0	0	2	0	2	45	47	0	0	2	0
8	8	A	27	3	55	5	2	0	0	2	0	0	3	3	0	0	0	0	0	0	0	0
9	9	A	5	68	5	0	0	0	3	0	18	0	0	0	0	0	0	0	0	0	0	0
10	10	A	5	2	0	0	7	3	8	0	0	0	0	0	0	3	58	2	0	3	3	5
11	11	A	65	0	33	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
12	12	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	65
13	13	A	0	95	0	3	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
14	14	A	0	0	0	0	0	0	0	7	3	0	38	37	0	0	2	3	0	3	3	3
15	15	A	0	0	0	0	0	0	0	2	8	0	17	30	0	0	5	8	0	10	12	8
16	16	A	0	3	0	2	0	0	2	45	2	0	2	0	0	0	18	10	2	12	3	0

SEQUENCE PROFILE AND ENTROPY

SeqNo	PDBNo		V	L	I	M	F	W	Y	G	A	P	S	T	C	H	R	K	Q	E	N	D
1	1	A	0	22	6	72	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	2	A	3	0	0	3	14	0	22	22	6	3	0	3	0	0	3	14	0	3	0	6
3	3	A	0	0	0	0	0	0	2	93	2	0	0	0	0	0	2	0	0	0	0	0
4	4	A	0	0	2	0	13	26	50	0	2	0	0	2	0	0	2	0	0	0	2	0
5	5	A	0	0	0	4	0	20	0	0	17	0	0	17	4	11	0	7	7	0	13	0
6	6	A	0	0	0	0	0	0	0	72	0	0	2	0	2	4	2	0	0	2	9	7
7	7	A	2	0	0	0	0	0	0	0	0	0	0	2	0	2	45	47	0	0	2	0
8	8	A	27	3	55	5	2	0	0	2	0	0	3	3	0	0	0	0	0	0	0	0
9	9	A	5	68	5	0	0	0	3	0	18	0	0	0	0	0	0	0	0	0	0	0
10	10	A	5	2	0	0	7	3	8	0	0	0	0	0	0	3	58	2	0	3	3	5
11	11	A	65	0	33	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
12	12	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	65
13	13	A	0	95	0	3	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
14	14	A	0	0	0	0	0	0	0	7	3	0	38	37	0	0	2	3	0	3	3	3
15	15	A	0	0	0	0	0	0	0	2	8	0	17	30	0	0	5	8	0	10	12	8
16	16	A	0	3	0	2	0	0	2	45	2	0	2	0	0	0	18	10	2	12	3	0

Figure 2.3 An Example of the sliding window technique with a widow size of 9 applied on 1b25 HSSP file

2.3 Distance Measure

According to [8, 15], the city block metric is more suitable for this field of study since it will consider every position of the frequency profile equally. The following formula is used to calculate the distance between two sequence segments [15]:

$$\text{Distance}=\sum_{i=1}^{L}\sum_{j=1}^{N}\left|F_k(i,j)-F_c(i,j)\right| \tag{2.1}$$

Where *L* is the window size and *N* is 20 which represent 20 different amino acids. *Fk(i,j)* is the value of the matrix at row *i* and column *j* used to represent the sequence segment. *Fc(i,j)* is the value of the matrix at row *i* and column *j* used to represent the centroid of a give sequence cluster.

2.4 Structural Similarity Measure

Cluster's average structure is calculated using the following formula:

$$\frac{\sum_{i=1}^{ws}\max(\,p_{i,H}\,,\,p_{i,E}\,,\,p_{i,C}\,)}{ws} \tag{2.2}$$

Where *ws* is the window size and *Pi,H* shows the frequency of occurrence of helix among the segments for the cluster in position *i*. *Pi,E* and *Pi,C* are defined in a similar way. If the structural homology for a cluster exceeds 70%, the cluster can be considered structurally identical [17]. If the structural homology for the cluster exceeds 60% and is lower than 70%, the cluster can be considered weakly structurally homologous [8].

2. 5 Davis-Bouldin Index (DBI) Measure

Besides using secondary structure information as a biological evaluation criterion, we also include a computer science point of view evaluation method for feature selection as part of our work. The DBI measure [19] is a function of the inter-cluster and intra-cluster distance. A good cluster result should reflect a relatively large inter-cluster distance and a relatively small intra-cluster distance. The DBI measure combines both distance information into one function, which is defined as follows:

$$\text{DBI}=\frac{1}{k}\sum_{p=1}^{k} \underset{p\neq q}{MAX}\left\{\frac{d_{\text{int}ra}(C_p)+d_{\text{int}ra}(C_q)}{d_{\text{int}er}(C_p,C_q)}\right\}, \text{ where} \qquad (2.3)$$

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

k is the total number of clusters, $d_{\text{int}ra}$ and $d_{\text{int}er}$ 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.

2. 6 HSSP-BLOSUM62 Measure

BLOSUM62 [20] (Figure 2.4) is a scoring matrix based on known alignments of diverse sequences.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0	-2	-1	0	-4
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3	-1	0	-1	-4
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3	3	0	-1	-4
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3	4	1	-1	-4
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1	-3	-3	-2	-4
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2	0	3	-1	-4
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2	1	4	-1	-4
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3	-1	-2	-1	-4
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3	0	0	-1	-4
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3	-3	-3	-1	-4
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1	-4	-3	-1	-4
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2	0	1	-1	-4
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1	-3	-1	-1	-4
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1	-3	-3	-1	-4
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2	-2	-1	-2	-4
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2	0	0	0	-4
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0	-1	-1	0	-4
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3	-4	-3	-2	-4
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1	-3	-2	-1	-4
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4	-3	-2	-1	-4
B	-2	-1	3	4	-3	0	1	-1	0	-3	-4	0	-3	-3	-2	0	-1	-4	-3	-3	4	1	-1	-4
Z	-1	0	0	1	-3	3	4	-2	0	-3	-3	1	-1	-3	-1	0	-1	-3	-2	-2	1	4	-1	-4
X	0	-1	-1	-1	-2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2	0	0	-2	-1	-1	-1	-1	-1	-4
*	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	1

Figure 2.4 BLOSUM62 Matrix

By using this matrix, we may access the consistency of the amino acids appearing in the same position of the motif information generated by our method. Because different amino acids appearing in the same position should be close to each other, the corresponding value in the BLOSUM62 matrix will give a positive value. Hence, the measure is defined as the following:

If k = 0 or 1 **Then:** HSSP-BLOSUM62 measure = 0 (2.4)

Else: HSSP-BLOSUM62 measure $= \dfrac{\sum_{i=1}^{k-1}\sum_{j=i+1}^{k} HSSP_i \cdot HSSP_j \cdot BLOSUM\ \ 62_{ij}}{\sum_{i=1}^{k-1}\sum_{j=i+1}^{k} HSSP_i \cdot HSSP_j}$

k is the number of amino acids with a frequency higher than a certain threshold in the same position (in this work, 8% is the threshold; Since there are 20 different amino acids, in the

statistical point of view, each amino acid shares 5% occurrence opportunity. Therefore, we believe 8% is a reasonable value). *HSSPi* indicates the percent of amino acid *i* to be appeared. *BLOSUM62ij* denotes the value of BLOSUM62 on amino acid *i* and *j*. The higher HSSP-BLOSUM62 value indicates more significant motif information. To the best of our knowledge, this is the first time that HSSP and BLOSUM62 are combined and used as an evaluation method.

The major purpose of Equation 2.4 is to give a numerical value to evaluate the interchangeability of the motifs appearing on the same location. Therefore, when no noticeable amino acid or merely one noticeable amino acid appears on one location, we assign zero value to the position since no other amino acid candidates can be exchanged. Nevertheless, if we treat HSSP-BLOSUM62 measure in motif quality evaluation point of view, while k equals one, it indicates that there is only one amino acid appearing in the position. Unlike Equation 2.4, we believe this situation should be assigned some positive value to account for the clear information. Therefore, we assign the corresponding amino acid's diagonal value ($BLOSUM62_{ii}$) in BLOSUM62. Thus, the measure is modified as the following:

If k = 0: HSSP-BLOSUM62 measure = 0 (2.5)

Else If k = 1: HSSP-BLOSUM62 measure = $BLOSUM62_{ii}$

Else: HSSP-BLOSUM62 measure = $$\frac{\sum_{i=1}^{k-1}\sum_{j=i+1}^{k} HSSP_i \cdot HSSP_j \cdot BLOSUM\ \ 62_{ij}}{\sum_{i=1}^{k-1}\sum_{j=i+1}^{k} HSSP_i \cdot HSSP_j}$$

After the modification, we notice that when k=1, for most of the time, the $HSSP_i$ value is high. However, under some circumstances, the $HSSP_i$ value is just marginally past the threshold (8%). If we assign $BLOSUM62_{ii}$ to both situations, it makes HSSP-BLOSUM62

measure unfair. As a result, we modify the measure once again and it becomes the following formula:

If k = 0: HSSP-BLOSUM62 measure = 0 (2.6)
Else If k = 1:
If $HSSP_i > 10\%$: HSSP-BLOSUM62 measure = $BLOSUM62_{ii}$
If $8\% \le HSSP_i < 10\%$: HSSP-BLOSUM62 measure = $\frac{1}{2} BLOSUM62_{ii}$
Else: HSSP-BLOSUM62 measure = $\frac{\sum_{i=1}^{k-1} \sum_{j=i+1}^{k} HSSP_i \cdot HSSP_j \cdot BLOSUM62_{ij}}{\sum_{i=1}^{k-1} \sum_{j=i+1}^{k} HSSP_i \cdot HSSP_j}$

Since the HSSP-BLOSUM62 is a new evaluation measure, we are still trying to improve it. In this work, we regard Equation 2.4 as HSSP-BLOSUM62 1.0 (the first version), Equation 2.5 as HSSP-BLOSUM62 2.0, and Equation 2.6 as HSSP-BLOSUM62 2.1.

CHAPTER3

FUZZY IMPROVED K-MEANS (FIK) GRANULE COMPUTING MODEL FOR PROTEIN SEQUENCE MOTIF DISCOVERING

3.1 Motivation

In order to generate higher quality protein sequence motif information from Zhong's [8] dataset, we tried several different advanced clustering algorithms, such as hierarchical clustering, SOM etc. However, since the dataset itself contains more than 560,000 segments where each segment contains 180 dimensions, any clustering algorithm required more than $O(n^2)$ complexity is not applicable. Therefore, the very first step of our research is trying to separate the whole data space into several smaller pieces. But deciding how to cut is an issue. If we just use the traditional clustering concept to cluster all data segments into clusters, the crisp cut work against the concept of finding motif information across family boundaries. As a result, we realized that we need to utilize the power of granular computing techniques. In our model, Fuzzy C-means clustering algorithm works as the first step to separate the whole dataset into several smaller informational granules, and then it is followed by applying advanced K-means clustering algorithm.

3.2 Granular Computing Techniques

Granular computing [21-31] represents information in the form of aggregates, also called "information granules." 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 meaningless information. As opposed to traditional data-oriented numeric computing, granular computing is knowledge-oriented [23]. Some formal models under the granular computing include:

- Set theory and interval analysis
- Fuzzy sets
- Rough sets
- Probabilistic sets
- Clusters

Since the dataset we deal with contains a large amount of information, granular computing is a very useful tool. Two models, fuzzy sets and clusters, are applied in our work.

3.2.1 K-means Clustering Algorithm

The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering [32]. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. In machine learning point of view, clustering is a typical example of unsupervised learning which indicates the process of learning does not rely on predefined classes. Therefore, clustering is a form of learning by observation, rather than learning by examples [32].

Among all clustering algorithms, K-means clustering algorithm has the advantages of easy interpretation and implementation, high scalability, and low computation complexity. The K-means clustering take the user input parameter k, and partitions a set of n objects into k clusters so that the resulting intra-cluster similarity is high but the inter-cluster similarity is low [32]. The pseudo code of basic K-means clustering algorithm is described in Figure 3.1:

(1) arbitrarily choose K objects as the initial centroid;
(2) **repeat**
(3) (re)assign each remaining object to the cluster to which the object is the most Similar, based on the mean value of the object in the cluster;
(4) update the cluster mean;
(5) **until** no change;

Figure 3.1 The K-means Clustering Algorithm

3.2.2 Fuzzy C-means (FCM) Clustering Algorithm

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 [33] and improved by Bezdek in 1981 [34]) is frequently used in pattern recognition. The main purpose of this algorithm is trying to minimize the following objective function:

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

where *m, the* fuzzification factor, is any real number greater than 1, *uij* is the degree of membership of *xi* in the cluster *j*, *xi* is the *i*th of d-dimensional measured data, *cj* is the d-dimension center of the cluster, and ||*|| 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 *uij* and cluster centers *cj* 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 $| U^{(k+1)} - U^{(k)} | < \varepsilon$, where ε is a termination criterion, and k are the iteration steps. This procedure converges to a local minimum or a saddle point of Jm. [35]. The algorithm is described as Figure 3.2:

(1) initialize membership function matrix, $U^{(0)}$, and randomly select a set of initial centroids.;
(2) **repeat**
(3) at k-step: update $U^{(k)}$ to $U^{(k+1)}$ by the function of u^{ij};
(4) calculate the centroid information by c^{j} function;
(5) **until** $| U^{(k+1)} - U^{(k)} | < \varepsilon$ [19];

Figure 3.2 The Fuzzy C-means Clustering Algorithm

3.3 Fuzzy Improved K-means (FIK) Model

3.3.1 Improved K-means Clustering Algorithm

To obtain a more global result, we collect five K-means results and then select the initial

centroids, which not only have 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 mentioned above, we do not check the distance with other initial seeds. However, due to the recurrently appearing centroids limitation, we may not collect all initial points by this method. We usually obtain one third to half the starting centers of the information granules required and get the other initial centroids randomly with a distance check: 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 between all existing centroids is greater than the threshold distance, this point will be included in the initialization array as a new centroid; otherwise, the new point is too close with existing centroids so that the new point should be discarded.

3.3.2 Combine FCM with Improved K-means Clustering Algorithm

A granular computing based learning model called "Fuzzy Improved K-means model" (FIK model) is proposed here. 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. Major advantages of FIK model are reduced time- and space- complexity, filtered outliers, and higher quality granular information results. At the first stage, all of the data segments 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. Figure 3.3 shows the sketch of the model.

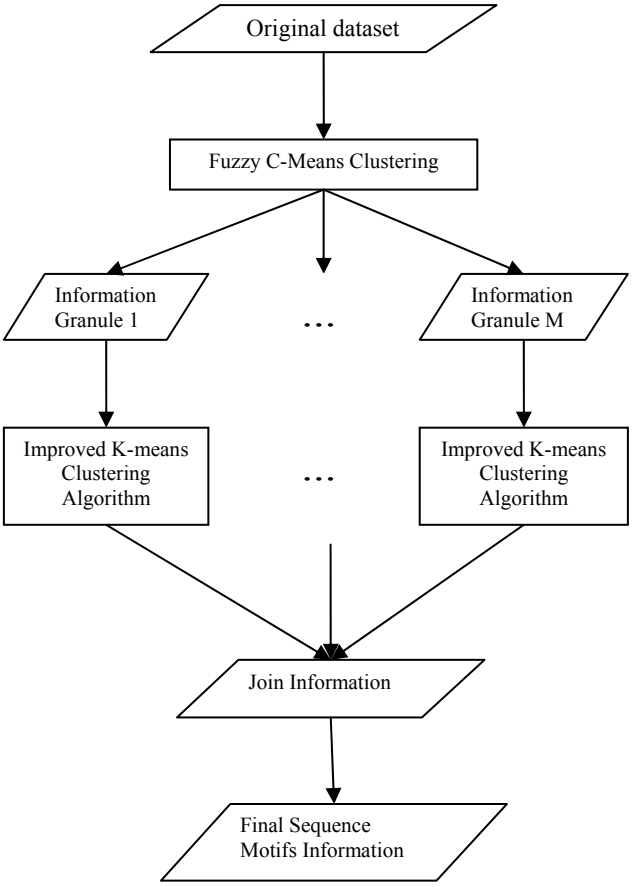


Figure 3.3 Fuzzy Improved K-means (FIK) model

3.4 Parameter Setup

In the previous work, Zhong *et al* [8] carefully chose 800 as the number of the clusters based on their experience and experiment. In order to compare with their results, we use the same number. In our work, 799 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. It is our best setup based on our trial-and-error results. Since our whole dataset includes more than 540,000 data

and each data contains 180 dimensions, the most common m value (m=2) is not suitable in our research. Even when m=1.1, the whole dataset cannot be separated because the membership value for every data equals close to 0.09 or 0.1. Therefore, we need to set m=1.05 to generate a crisp result to successfully separate the whole dataset. The reason we set the number of clusters for FCM as 10 is because of the physical limitation of computers. We tried to set the cluster as 15 or 20, but under these conditions, the fuzzifier value need to further decrease in order to have an identifiable membership value. When the fuzzifier value equals 1.05, it indicates that every number needs a power of twenty operations. If we further decrease the value of "m," overflow occurs. In order to separate information granules from FCM results, the membership threshold is set to 13%. Using this value, we filter out around 15% as outliers from the dataset and assign the rest of the data to one or more clusters. Since we divide whole dataset into 10 smaller information granules, the lowest threshold should be 10%. We tried the threshold from 11% to 15% and realized that 13% is the most suitable one; otherwise, each information granule contains too many or too few members. The function that decides how many numbers of clusters should be in each information granule is given bellow:

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

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. Table3.1 summarizes 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.

Table 3.1 Summary of results obtained by 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	143	94.6MB
Granule 7	4583	5	3.3MB
Granule 8	43254	48	31.7MB
Granule 9	5032	6	3.7MB
Total	**721390**	**799**	**529MB**
Original dataset	**562745**	**800**	**413MB**

3.5 Experimental Results

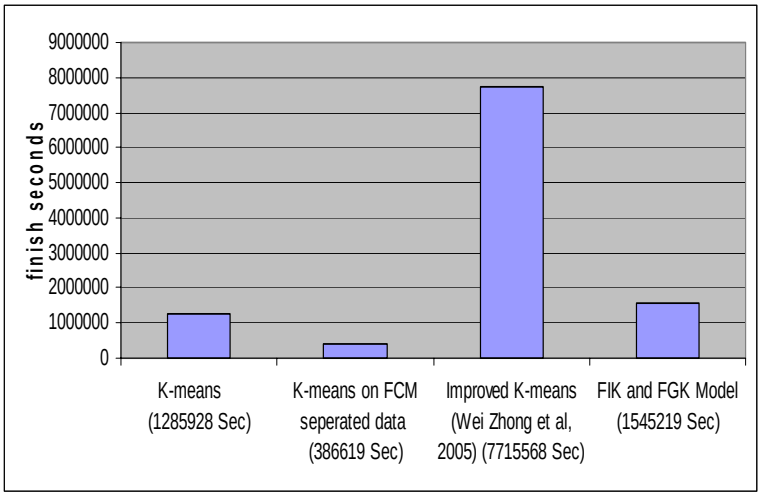
3.5.1 Comparison of Execution Time

In Table 3.2, 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 chapter is shown. From the graph, “K-means” represents the average execution time for applying the original K-means algorithm on the intact dataset once. “K-means on FCM separated data” gives the average run time for executing the 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 informational granules plus the time required by Fuzzy Cmeans clustering algorithm (154899 seconds). The

third method, “Improved K-means” created by Zhong et al [8] 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, their method requires six iterations of K-means clustering algorithm on the original dataset. Therefore, the value shown on the graph equals the “K-means” value times six. The last method, “FIK and FGK model,” is the model presented in this chapter. The method to compute the total required time is similar to Zhong’s method: the sum of execution time on all information granules times six plus one iteration time required by FCM.

Table 3.2 Execution time comparison table

	Running Time (Sec)
Granul0	60546
Granul1	15198
Granul2	23603
Granul3	13680
Granul4	12771
Granul5	46230
Granul6	53430
Granul7	83
Granul8	6072
Granul9	107
Total	231720
Original dataset	1285928



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

3.5.2 Comparison of Protein Sequence Motif Quality

In Table 3.3, the novel HSSP-BLOSUM62 1.0 measure and average percentage of sequence segments belonging to clusters with high structural similarity for different methods is given. All numbers and standard deviation are obtained from five runs of each setting. The first column shows the different methods with different parameters. "Traditional" refers to the original K-means algorithm applied to the whole dataset. "FCM-Kmeans" 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. "FIK model 0" indicates the initial centroids' location generated by improved mechanism are the same with all other FIK models but there is no distance check criteria for those generated at random. Figure 3.4 to 6 are interpreted from Table 3.3.

Table 3.3 Comparison of HSSP-BLOSUM62 measure and percentage of sequence segments belonging to clusters with high structural similarity.

Different Methods	>60%	S.D.	>70%	S.D.	H-B Measure
Traditional	25.82%	0.93	10.44%	0.61	0.2543
FCM-K-means	37.14%	1.46	12.99%	0.74	0.3589
FIK Model					
FIK Model 0	40.15%	1.09	13.44%	0.49	0.3730
FIK Model 800	40.23%	0.45	13.37%	0.58	0.3717
FIK Model 1000	39.15%	0.39	13.27%	0.29	0.3665
FIK Model 1200	38.90%	0.43	12.89%	0.77	0.3697
FIK Model 1400	37.80%	0.80	12.59%	0.44	0.3655

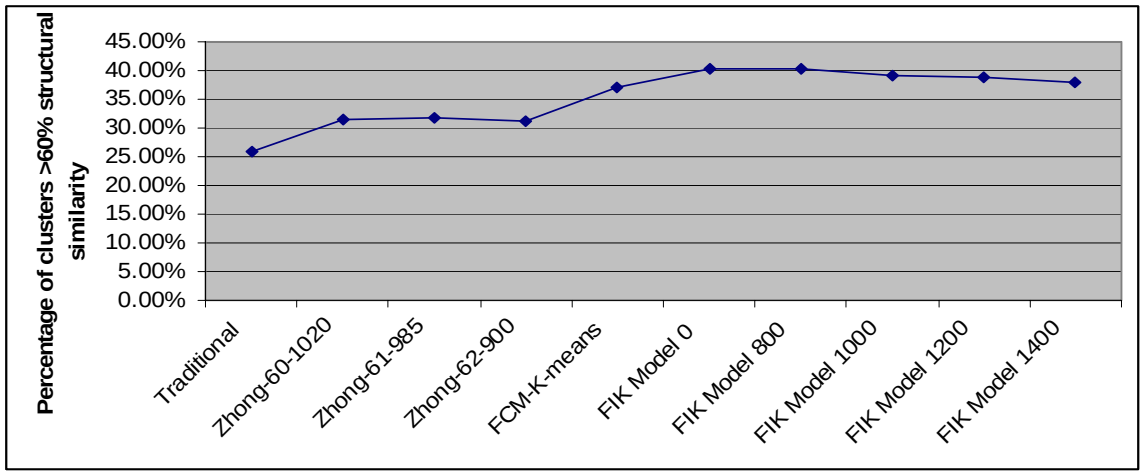


Figure 3.4 Comparison of percentage of sequence segments belonging to cluster with structure similarity greater than 60%

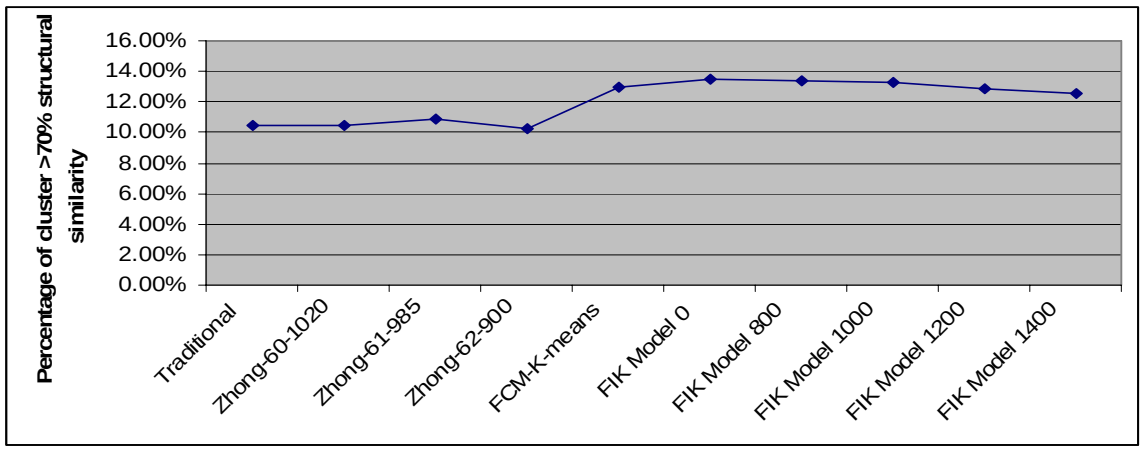


Figure 3.5 Comparison of percentage of sequence segments belonging to cluster with structure similarity greater than70%

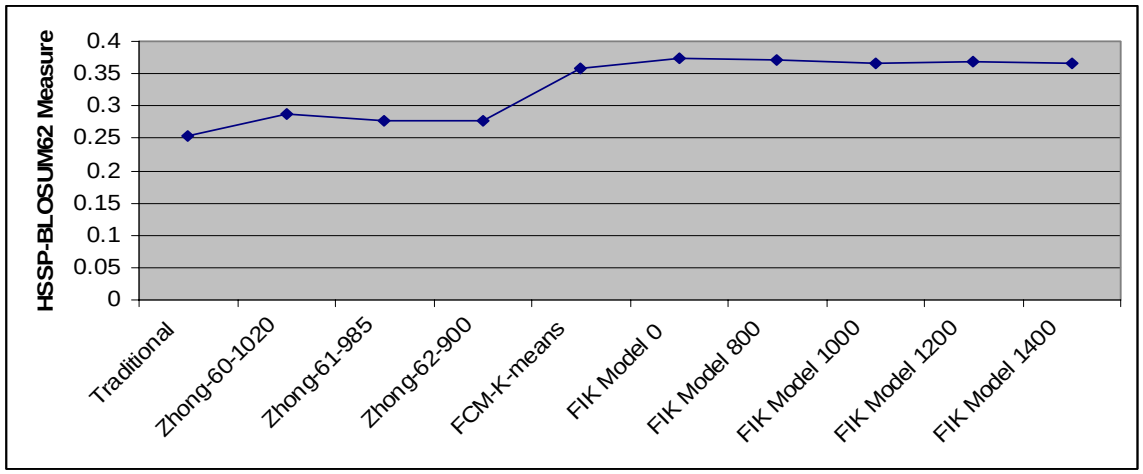


Figure 3.6 Comparison of the HSSP-BLOSUM62 1.0 measure

The results of Table 3.3 and Figures 3.4, 3.5 and 3.6 reveal that the quality of clusters improved dramatically by applying granular computing techniques which utilize FCM to separate the whole dataset into several information granules. The results of FCM-K-means indicates that the average percentage of clusters with structural similarity greater than 60% increased more than eleven percent, which translates to more than 90 meaningful sequence motifs that cannot be found by traditional methods but are discovered by our granular approach. Also, the HSSP-BLOSUM62 measurement increasing from 0.254 to 0.359 indicates that the motif information is more consistent and meaningful. Compared with the earlier work [8], we improved the structural similarity of clusters more than 10% in all models, while their best work increased only from 25.82% to 31.71%.

For the FIK model, the improved K-means clustering algorithm also plays an important role. Although the percentage of structural similarity greater than 60% increases only two to three percent of the structural similarity by comparing with FCM-K-means, 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. “FIK Model 0” and “FIK Model 800” perform better using the same model, the structural similarities as well as HSSP-BLOSUM62 measure are all above average. However, due to omitting the threshold check while choosing the new random centroids, the quality of the clusters is not as stable as model 800. This can be seen using standard deviation provided in Table 4. The percentage of clusters with structural similarity higher than 70% is slightly improved in the first three FIK models. However, once the distance threshold is set too high, the clustering results suffer from both wasting too much time in choosing centroids and

choosing outliers as initial centroids, “FIK model 1400” is a good example of this. For small information granules (granule 7 and granule 9), Zhong et al could not find initial candidates under the 1400 distance measure threshold; we applied 1350 instead.

CHAPTER 4

FUZZY GREEDY K-MEANS (FGK) GRANULE COMPUTING MODEL FOR PROTEIN SEQUENCE MOTIF DISCOVERING

4.1 Fuzzy Greedy K-means (FGK) Model

4.1.1 Motivation

After we built up the Fuzzy Improved K-means (FIK) model, we tried to further improve the quality of protein sequence motif information. By carefully inspecting the greedy K-means clustering algorithm presented by Zhong [8], we realized that the greedy approach may have a good chance of producing a higher structural similarity result. After testing several different approaches, we provide a new greedy K-means clustering algorithm and apply it in our granule computing model.

4.1.2 Zhong's Improved K-means Clustering Algorithm

This method is proposed by Zhong et al [8] 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 that method, the original random Kmeans clustering algorithm was performed five times. In each round, initial points that have the potential to form a cluster with high structural similarity are chosen for the improved Kmeans clustering algorithm. Each time a new potential initial center is chosen, its distance is checked against all points that have already been 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.

4.1.3 New Greedy K-means Clustering Algorithm

Our greedy initialization method for K-means clustering is similar to the method of Zhong *et al* [8], but greedier. Instead of randomly picking initial seeds in each round of the original Kmeans, we collect all five K-means results and then select the initial centroids. Due to the fact that the centroids in higher quality clusters have the potential to generate better clusters in the sixth round, we divide our selection procedure into five steps: initially gathering centroid seeds belonging to clusters with structural similarity greater than 80% and then proceeding with 75%, 70%, 65% and 60%. The minimum distance strategy mentioned in Zhong's approach also applies to this method. Results with different distance thresholds are given in section four. Compared with our improved K-means algorithm mentioned in section 2-C, this method can gather more initial seeds. If we set the minimum distance measurement to 250 while gathering initial seeds for the sixth round, we can always obtain many more centroids than the number we need. Therefore, in this case, we only collect initial seeds until the amount is met and discard the rest. However, if the distance measurement threshold is set to 350, sometimes the number of initial centroids acquired is not enough. In this case, we use a random method with the minimum distance 800 to choose the rest of required centroids.

4.1.4 Combine FCM with New Greedy K-means Clustering Algorithm

Basically, the Fuzzy Greedy K-means granule computing model is similar to Fuzzy Improved K-means model we mentioned in previous chapter. All data segments are also clustered by Fuzzy C-Means into several "functional granules" as the first step. For each granule, the new greedy K-means clustering algorithm is applied instead of using the improved K-means clustering algorithm. In the end, we join the information generated by all granules and obtain the final sequence motifs information. Figure 4.1 shows the sketch of the model.

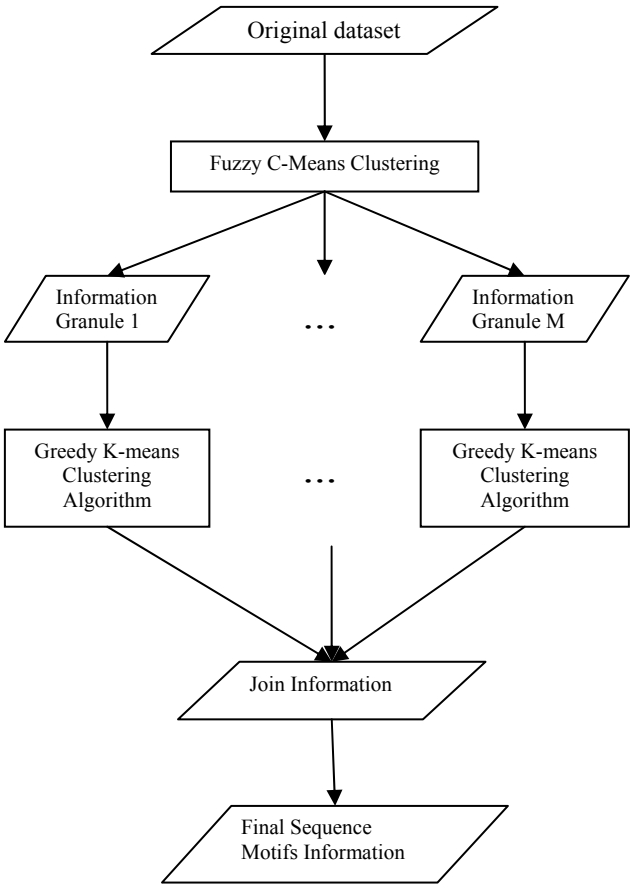


Figure 4.1 Fuzzy Greedy K-means (FGK) model

4.2 Experimental Results

All results mentioned in this section are generated from the same parameters we used in the previous chapter. Since FIK and FGK models have almost the same execution time of selecting initial centroid location after five traditional K-means clustering are performed, the results of execution time comparison is similar. Therefore, we do duplicate the results here.

The major difference (improvement) happens on the evaluation measure of secondary structural similarity as we expected. In Table 4, the novel HSSP-BLOSUM62 measure and average percentage of sequence segments belonging to clusters with high structural similarity for different methods is given. To fully compare these results with the results we mentioned in the last chapter, we include the traditional K-means result, the Fuzzy K-means result and the selected best result generated by FIK model shown in chapter 3. Also, in order to fully compare with the latest results, we carried out Zhong's method in our dataset. Due to the difference of dataset and window size between ours and [8], we cannot set exactly the same parameters to obtain the comparable results. However, according to the conclusion in [8], a higher minimum distance limitation among initial centroids may yield better quality results. Therefore, we start to simulate their results by collecting initial points which have the ability to generate clusters with structure similarity higher than 60% and maximize the minimum distance check threshold. "Zhong-60-1020" indicates that we collect the initial points that generate clusters have higher than 60% structural similarity and the minimum distance check threshold equals 1020. If we set the distance threshold higher than 1020, we cannot gather all 800 initial centroids from five iterations of traditional K-means. Since all improved K-means methods we obtained in this work are based on five runs of original K-means, we believe the comparison is fair. In addition, the improvement is similar to the results of [8]. "Zhong-61-985" and "Zhong-62-900" are defined in

same manner. If we set the structural similarity threshold up to 63%, the total number of qualified clusters is already less than 800. As a result, "Zhong-63-distance" or higher can not be performed.

"FGK model 200" indicates that the dataset is clustered by the FGK model with the new greedy initialization K-means clustering algorithm, and the distance threshold is set as 200. "FGK model 250," "FGK model 300," "FGK model 350," and "FGK model 400" are defined similarly. Figure 4.2 to 4.4 are interpreted from Table 4.1. In order to generate a more manageable view of our large set of results, we select "Zhong-61-985," "FIK Model 800," and "FGK Model 250" as representatives for Zhong's simulation, FIK Model and FGK Model results, respectively.

Table 4.1 Comparison of HSSP-BLOSUM62 measure and percentage of sequence segments belonging to clusters with high structural similarity.

Different Methods	**>60%**	**S.D.**	**>70%**	**S.D.**	**H-B Measure**
Traditional	25.82%	0.93	10.44%	0.61	0.2543
Zhong-60-1020	31.46%	0.26	10.42%	0.59	0.2871
Zhong-61-985	31.71%	0.81	10.84%	0.07	0.2784
Zhong-62-900	31.04%	0.19	10.29%	0.64	0.2768
FCM-K-means	37.14%	1.46	12.99%	0.74	0.3589
FIK Model 800	40.23%	0.45	13.37%	0.58	0.3717
FGK Model					
FGK Model 200	42.45%	0.06	14.14%	0.02	0.3393
FGK Model 250	42.77%	0.07	14.06%	0.07	0.3443
FGK Model 300	41.08%	0.14	13.89%	0.02	0.3311
FGK Model 350	37.47%	0.51	13.49%	0.14	0.3489
FGK Model 400	37.62%	1.56	13.86%	1.29	0.3676
Best Selection	44.18%	0	15.02%	0	0.3664

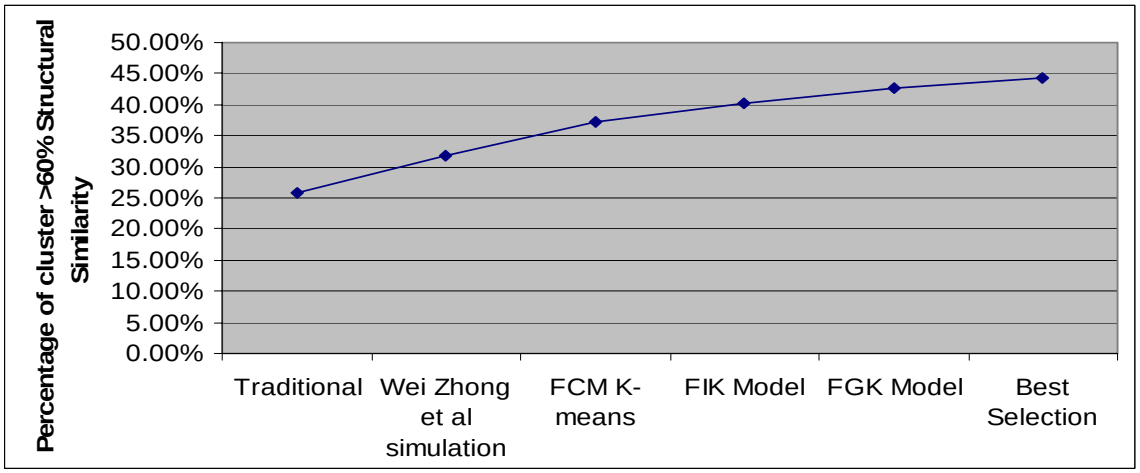


Figure 4.2 Comparison of percentage of sequence segments belonging to cluster with structure similarity greater than 60%

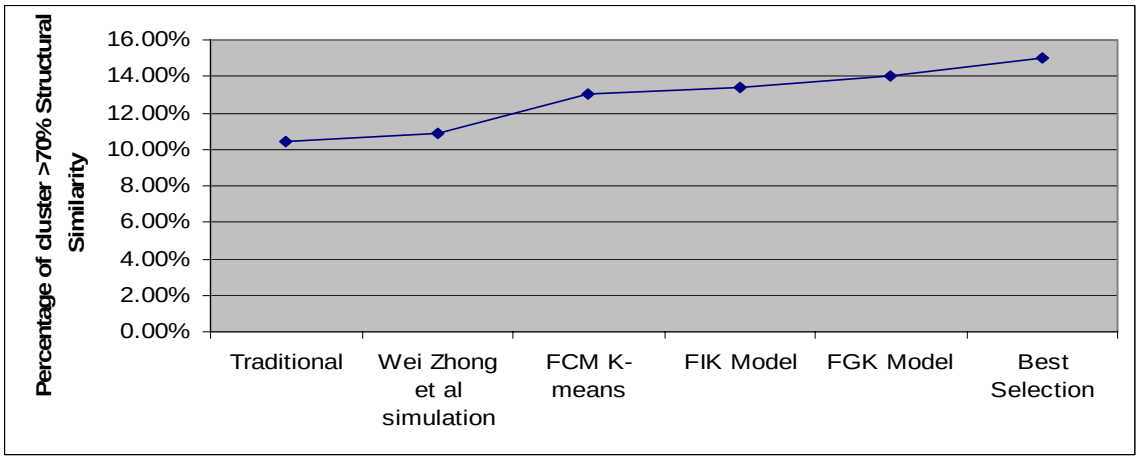


Figure 4.3 Comparison of percentage of sequence segments belonging to cluster with structure similarity greater than 70%

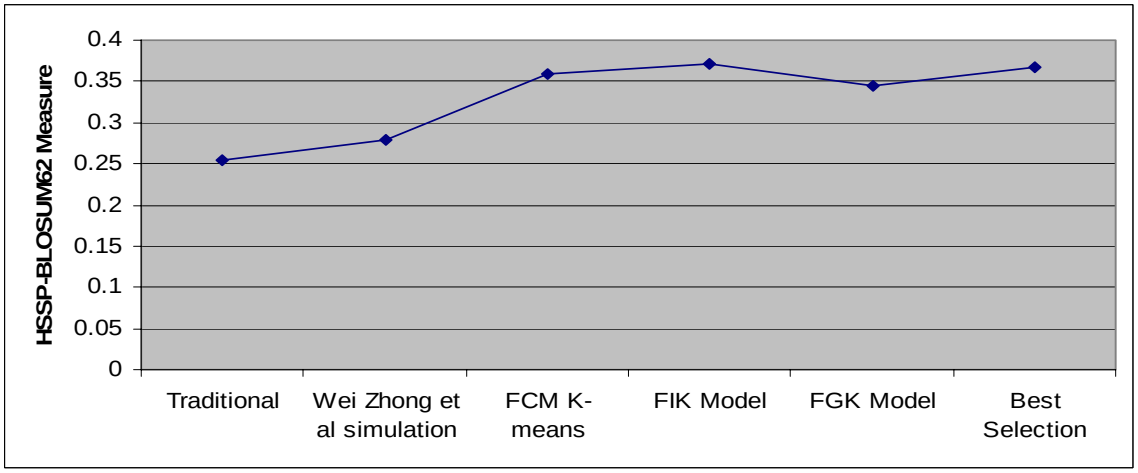


Figure 4.4 Comparison of the HSSP-BLOSUM62 measure

The quality of cluster obtained by Zhong's method increased around five to six percent in structural similarity greater than 60%, which matches their results. Compared with their work, we improved the structural similarity of clusters more than 10% in all models, while their best work increased only from 25.82% to 31.71%.

For "FGK model 250," although the measurement of HSSPBLOSUM62 decreased slightly, the result achieves the highest percentage (42.77%) of clusters with high structural similarity among all methods. It indicates that greedy initialization method can reveal some hidden motif information that the traditional one can not. By comparing the result that was generated by the FIK model and the FGK model, we realized that the FGK model has the ability to focus on one specific measurement and improve the value. On the other hand, the FIK model is more suitable for a global view and increases all measures evenly. This aside, both models did a great job in improving the quality of clusters from the traditional K-means approach.

While further examining the results in each information granule generated by the FIK and the FGK models, we discovered that sometimes the FGK model generates better results and sometimes the FIK performs better. This may be due to each information granule having different characteristics. Adjusting parameters may capture the best result. Since each information granule is independent of each other, we collected the best clustering results in each granule from FGK-250, FIK-00, and FIK-800 and generate the "Best Selection" as our representative sequence motif information in this work. Not surprisingly, this produces the best results in the structural similarity and above average results in the HSSP-BLOSUM62 measurement.

4.3 Protein Sequence Motifs

After discovering more than 300 high quality sequence motifs, how to present these useful recurring patterns is another research issue. In this section, we present two formats to describe our discovery. We first show the format used by Zhong et al and our early stage research. And then we demonstrate a new format which combines amino acid logos for presenting more detailed information.

4.3.1 Old Presentation Format

The Table 4.4 to 4.9 illustrates eight different sequence motifs generated from our "best selection." Due to space limitation, we only present part of our recurring pattern information in this work. The following format is used for the representation of each motif table.

- The first row represents the number of members belonging to this motif, the secondary structural similarity and the average HSSP-BLOSUM62 value.
- 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 appearing in the given position. If the amino acids are appearing with the frequency higher than 10%, they are indicated by upper case; If the amino acids are 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 fourth column indicates the value of the HSSP-BLOSUM62 measure.

- The last column indicates the representative secondary structure to the position.

Table 4.2 Helices motif with conserved A K E

Number of segments: 969
Structure homology: 83.12%
AvgHSSP-BLOSUM62: 0.375

#	Noticeable Amino Acid	H	B	S
1	a K E D	.25	.07	H
2	A K E D	.24	.04	H
3	l a	.43	-1.0	H
4	V L I	.63	1.9	H
5	A K E D	.21	.01	H
6	a R K E	.32	.05	H
7	L I	.86	2.0	H
8	a R K E	.25	.36	H
9	A r K E	.24	-.08	H

Table 4.3 Helices motif with conserved A

Number of segments:1606
Structure homology: 81.17%
AvgHSSP-BLOSUM62: 0.044

#	Noticeabl Amino Acid	H	B	S
1	A E D	.28	-.15	H
2	A d	.41	-.20	H
3	V L I a	.69	1.0	H
4	A r K E	.33	-.02	H
5	A K q E d	.26	.03	H
6	A	.83	0	H
7	V L I	.82	1.9	H
8	A r K E d	.25	-.15	H
9	A r K E	.35	-.23	H

Table 4.4 Helices motif with conserved either A or L

Number of segments: 2017
Structure homology: 73.40%
AvgHSSP-BLOSUM62: -.059

#	Noticeable Amino Acid	H	B	S
1	A e d	.32	-.55	H
2	L I	.93	2.0	H
3	L r	.43	-2.0	H
4	ArKqEd	.22	.09	H
5	I A	.44	-1.0	H
6	L i	.93	2.0	H
7	a r K e	.30	.09	H
8	a k E d	.25	-.15	H
9	I A	.39	-1.0	H

Table 4.5 Helices-Coil motif

Number of segments:1870
Structure homology: 74.05%
AvgHSSP-BLOSUM62: 0.123

#	Noticeabl Amino Acid	H	B	S
1	V L I A	.58	.48	H
2	A r K E d	.21	-.10	H
3	I A r e	.43	-.13	H
4	L i	.87	2.0	H
5	A r K e	.34	.02	H
6	A K E d	.22	-1.2	H
7	L A	.39	-1.0	C
8	G	.04	0	C
9	V L I f	.65	1.1	C

Table 4.6 Hydrophobic Coil motif with conserved G A S T

Number of segments: 620
Structure homology: 61.33%
AvgHSSP-BLOSUM62: -.018

#	Noticeable Amino Acid	H	B	S
1	G A s t	.32	-.08	C
2	G a S t	.28	-.04	C
3	G A S T	.31	-.01	C
4	g A s T	.41	.05	C
5	v A s T	.41	.08	C
6	G a s t	.39	-.03	C
7	G A S t	.34	-.02	C
8	G A S t	.29	-.08	C
9	G A S n	.27	.01	C

Table 4.7 Coil-sheet-coil motif with conserved VLI in E

Number of segments:628
Structure homology: 63.66%
AvgHSSP-BLOSUM62: 0.747

#	Noticeabl Amino Acid	H	B	S
1	P s e d	.29	-.21	C
2	G E n D	.21	.15	C
3	G e d	.23	-.67	C
4	R K e	.22	1.1	C
5	V L I	.84	1.8	E
6	V L I	.75	1.9	E
7	V L I a	.55	.77	E
8	V L I	.57	1.9	E
9	A S t e D	.26	-.01	C

Table 4.8 Sheet-Coil-sheet motif with conserved VLI in E

Number of segments: 854
Structure homology: 66.16%
AvgHSSP-BLOSUM62: 0.980

#	Noticeable Amino Acid	H	B	S
1	V L I	.59	2.0	E
2	V L I	.69	2.2	E
3	v a	.39	2.0	E
4	V L i	.51	1.1	E
5	a K E	.30	1.9	C
6	g e N D	.13	-.29	C
7	G	.07	.67	C
8	r K E	.21	-.64	C
9	V L I	.67	-.28	E

Table 4.9 Helices-Coil-sheet motif

Number of segments: 1475
Structure homology: 73.91%
AvgHSSP-BLOSUM62: 0.047

#	Noticeabl Amino Acid	H	B	S
1	L i A	.57	-.13	H
2	L A	.77	-1.0	H
3	A R K E	.29	.15	H
4	A r K E d	.23	-.12	H
5	L A	.56	-1.0	C
6	G	.04	0	C
7	V L I A	.64	.64	C
8	P K E D	.27	-.09	C
9	V L I	.74	1.9	E

4.3.2 Novel Presentation Format

Based on the format we showed in 4.3.1, we propose a new motif information representation which utilizes graphical amino acid logos that is widely used by biologists. It improves the previous used frequency table by showing the noticeable amino acids clearly together with the frequency scale. Thus, the protein sequence motifs transcending protein families discovered and extracted by computer scientists can be easily understood by biologists. The Table 4.9 ~4.14 illustrates some sequence motifs generated from our "best selection." The following format is used for representation of each motif table.

- The upper box gives the number of members belonging to this motif, the secondary structural similarity and the average HSSP-BLOSUM62 value.
- The graph demonstrates the type of amino acid frequently appearing in the given position by amino acid logo. It only shows the amino acid appearing with a frequency higher than 8%. The height of symbols within the stack indicates the relative frequency of each amino or nucleic acid at that position.

- The x-axis label indicates the representative secondary structure (S), the HSSP-BLOSUM62 measure (H-B) and the hydrophobicity value (Hyd.) of the position. The hydrophobicity value is calculated from the summation of the frequencies of occurrence of Leu, Pro, Met, Trp, Ala, Val, Phe, and Ile.

Table 4.10
Helices motif with conserved A K E

Number of segments: 969
Structure homology: 83.12%
AvgHSSP-BLOSUM62: 0.375

S:	H	H	H	H	H	H	H	H	H
H-B	0.07	0.04	-1.00	1.93	0.01	0.05	2.00	0.36	-0.08
Hyd	0.25	0.24	0.43	0.63	0.21	0.32	0.86	0.25	0.24

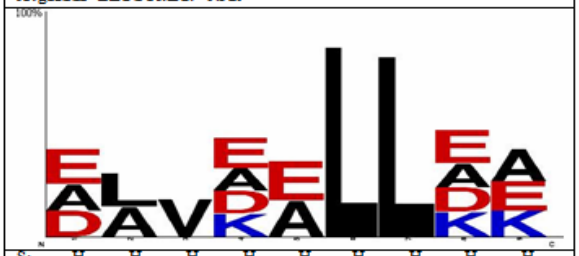
Table 4.11
Helices motif with conserved A

Number of segments: 1606
Structure homology: 81.17%
AvgHSSP-BLOSUM62: 0.044

S:	H	H	H	H	H	H	H	H	H
H-B	-0.15	-0.20	1.04	-0.02	0.03	0.00	1.87	-0.15	-0.23
Hyd	0.28	0.41	0.69	0.33	0.26	0.82	0.82	0.25	0.35

Table 4.12
Helices motif with conserved A or L

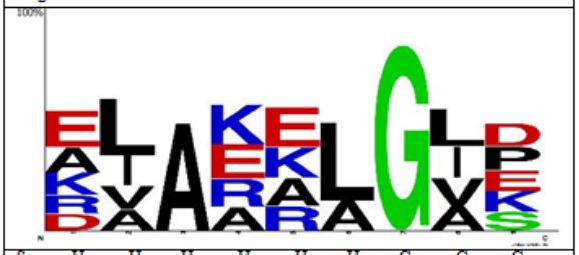
Number of segments: 1308
Structure homology: 63.34%
AvgHSSP-BLOSUM62: -0.329



S:	H	H	H	H	H	H	H	H	H
H-B	-0.16	-1.00	0.00	-0.22	-1.00	0.00	0.00	-0.19	-0.39
Hyd	0.32	0.44	0.45	0.31	0.29	0.94	0.94	0.28	0.33

Table 4.13
Helices-Coli motif

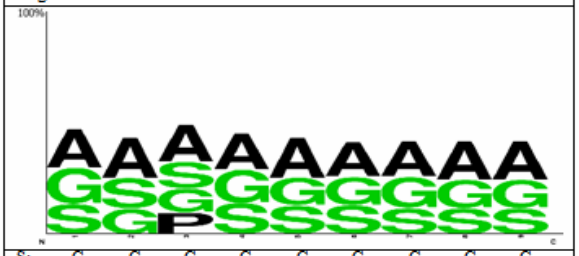
Number of segments: 1374
Structure homology: 79.10%
AvgHSSP-BLOSUM62: 0.056



S:	H	H	H	H	H	H	C	C	C
H-B	-0.16	0.83	0.00	0.22	0.02	-1.00	0.00	0.82	-0.22
Hyd	0.25	0.70	0.65	0.23	0.26	0.58	0.06	0.66	0.30

Table 4.14
Coli motif with conserved A G S

Number of segments: 1071
Structure homology: 64.67%
AvgHSSP-BLOSUM62: 0.275



S:	C	C	C	C	C	C	C	C	C
H-B	0.32	0.42	-0.32	0.34	0.36	0.35	0.32	0.35	0.34
Hyd	0.34	0.39	0.39	0.33	0.38	0.35	0.35	0.39	0.39

Table 4.15
Coil-sheet-coil motif with conserved VLI in E

Number of segments: 887
Structure homology: 65.97%
AvgHSSP-BLOSUM62: 0.761

S:	C	C	C	E	E	E	E	E	C
H-B	-1.0	0.00	2.00	0.77	2.26	2.17	1.01	0.06	-0.40
Hyd	0.26	0.58	0.07	0.63	0.87	0.78	0.65	0.41	0.33

Due to the space limitation, we only present six high quality motifs in this section. Nevertheless, Chapter 6 constructs a Super-Rule-Tree on all motifs with more than 60% secondary structural similarity; more detail motif information will be available in that chapter. Since it is clear that the new presentation format surpasses the old one, all motif information presented in the following chapters adapts the novel format.

4.4 Conclusion

In chapter 3 and chapter 4, two novel granular computing models that combine Fuzzy C-means and Improved K-means clustering algorithms have been proposed to solve high computational cost problems. In these models, we utilize fuzzy clustering to split the whole dataset into several information granules and analyze each granule by the K-means clustering algorithm with more advanced methods of initializing centroids. Analysis of sequence motifs also shows that the granular computing technology may detect some subtle sequence information overlooked by the K-means clustering algorithm alone. This is the first time the granular computing concept has been introduced using such a large, biologically meaningful dataset. Also, a novel biochemical measure, which combines the merits of the HSSP profile and the BLOSUM62 matrix, is proposed during this stage of the research. Compared with the latest results, our FIK and FGK models are capable of decreasing time and space complexity, filtering outliers, and capturing better results; the execution time is only 20% of the latest work and the quality of motif information is much better. Additionally, a new motif representation format which uses the graphical view of amino acid logo is shown in this work. We believe this new format provides

much more information, and our novel models are very powerful tools for bioinformatics research involving an extremely large database.

CHAPTER 5

EFFICIENT SUPER GRANULAR SVM FEATURE ELIMINATION (SUPER GSVM-FE) MODEL FOR PROTEIN SEQUENCE MOTIF INFORMATION EXTRACTION

5.1 Motivation

Feature selection is always closely related to machine learning techniques including supervised and unsupervised learning. For a dataset with lots of features, some of those may be useless or even harmful to the process of learning. A data maybe noisy itself, or maybe it will relate with some other data to cause a worse situation, such as confusing the mined information or hiding the impact caused by the true value.

The original dataset we work on contains more than 560,000 data segments. We believe that not all of them are very meaningful to the process of finding protein sequence motif information. There are two major reasons to support our claim: First, the information we try to generate is about sequence motif, which means only parts of sequences are useful to the clustering process. But the original input data are derived from whole protein sequences by the sliding window technique. This method considers every segment of the whole protein sequence as a candidate of being a motif. Although during the process of FIK and FGK models this process considers around 15% of segments as outliers, which is one of the reasons that our results improved, we still believe there are some segments need to be removed. Second, during fuzzy c-means clustering, it has the ability to assign one segment to more than one information granule.

However, not all data segments have direct relation to all of the granules to which they are assigned. Therefore, in this chapter, we tried to eliminate some segments in each information granule generated by FGK model in order to extract protein sequence motif information.

5.2 Support Vector Machine

Unlike clustering, Support Vector Machine is a supervised learning method, which learns by examples. A special property of SVM is that they simultaneously minimize the empirical classification error and maximize the geometric margin; hence they are also known as maximum margin classifiers.

There are three major different variations of SVM: classification SVM, regression SVM, ranking SVM and Multi-class SVM. The major difference between these models can be easily understood by their input target value, which is the most important value for the learning process. In classification mode, the target value denotes the class of the example. +1 as the target value marks a positive example, -1 a negative example respectively. In regression mode, the target contains the real-valued target value. In ranking mode, the target value is used to generate pair wise preference constraints. In multi-class mode, the target value is very similar to classification; the difference is that instead of using +1 or -1, multi-class SVM has the ability to have more than two target values.

Support vector machine has been wildly implemented since it was published. Among those different support vector machines, we realized that ranking SVM is the most suitable one to help us filter redundant segments. We set the target value for each member in the cluster by counting the number of matching secondary structure between member's structure and the representative

structure of the cluster. Since we use window size nine in our experiment, the highest target value is 9 and the lowest is 0. In this way, we don't have to tell the support vector machine whether we like or dislike (binary category) the feature. Instead, we give SVM our preference level, and let ranking SVM to tell us the rank of all segments.

5.3 Super Granule SVM Feature Elimination (Super GSVM-FE) Model for Motif Information Extraction

5.3.1 Super GSVM-FE Model

Basically, this new model is the next generation of FGK model. It also use fuzzy concept to divide original dataset into several smaller information granules. For each granule, after five iterations of traditional K-means clustering, the greedy k-means is applied. The next step is different from FGK model: we adapt ranking SVM to rank all members in each cluster generated by greedy K-means clustering algorithm, and then we filter out lower ranked members. The number of segments eliminated is decided by a user defined filtrate percentage. The results of different percentages are discussed in section four. After the feature elimination step, we collect all surviving data points in each information granule and then run greedy K-means with same initial centroids we previously generated. Finally, we collect all results in all granules to create final protein sequence motif information. Figure 12 is the sketch of the Super GSVM-FE model.

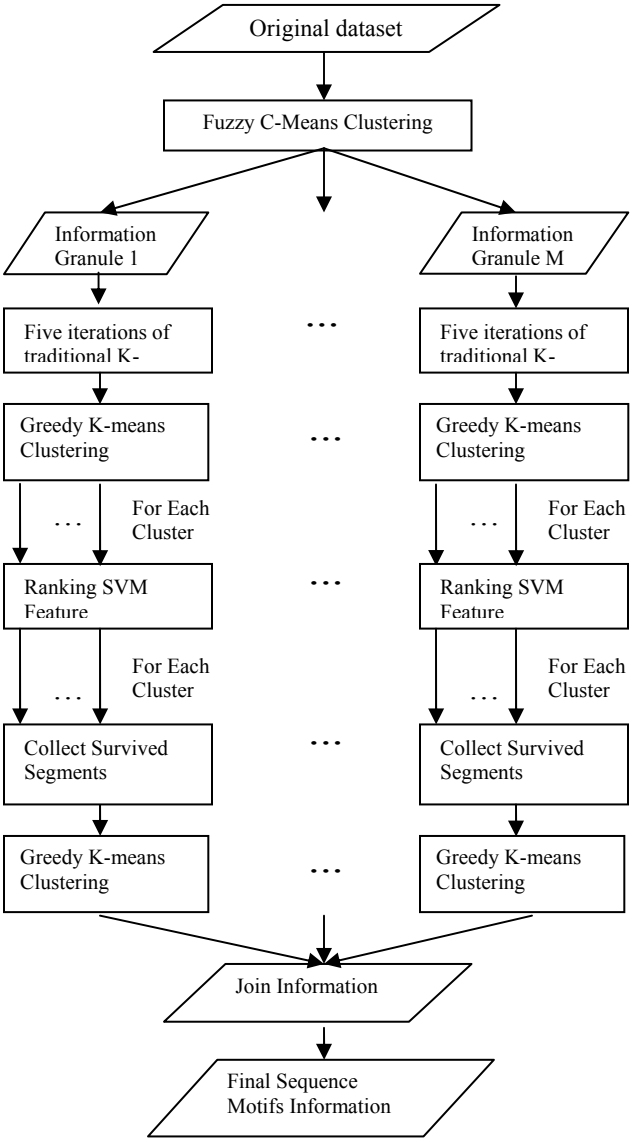


Figure 5.1 The Sketch of the Super GSVM-FE Model

In [25], a granular feature elimination model applied on Microarray data called GSVM-RFE is proposed. Although Microarray has the potential to deal with tens of thousands of genes simultaneously, compared with our data size, we have a much larger dataset. In their experiment, four clusters are divided. However, 800 clusters must be mined in our research. In addition, we use greedy K-means cluster algorithm to fix the initial centroid

location for generating clusters with higher quality and numerical stability. Therefore, we called our model super SVM-FE to indicate that our model has the potential to be applied to one huge data space.

5.3.2 Super Granule Shrink Feature Elimination Model

In order to compare the results, we present another similar feature elimination approach by modifying only one component of the model: we utilize shrink cluster size instead of ranking SVM. The number of segments eliminated is decided by a user defined distance threshold. If the distance between the member and the center of the cluster is greater than the threshold, the data point is filtered. The major advantage of this approach is that not all clusters get rid of the same amount members. If the cluster is compact at the beginning, fewer members should be eliminated. On the other hand, if the cluster is in a loose form, more data points should be obviated. The results of different thresholds are also discussed and compared in section four.

5.3.3 Experimental Dataset

Due to time limitations we carry out a complete process of our new approach on information granule number eight (as showed in Figure 5.2) which contains 43254 data segments and 48 clusters. After five iterations of traditional K-means clustering are executed, 45 initial centroids are decided for greedy K-means. All initial centers have at least a 250 distance measure from existing centroids. Another three initial seeds are generated randomly with a minimum distance threshold check.

	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 ← Testing DATA
Granule 9	5031	6	3.7MB
Total	**721390**	**799**	**529MB**
Original dataset	**562745**	**800**	**413MB**

Figure 5.2 The experimental dataset tested by the Super GSVM-FE model

5. 4 Results

5.4.1 Training Ranking-SVM Execution Time Comparison

Since the training step of Ranking-SVM is very time consuming, we want to minimize it. Due to the fact that the training time of SVM increases exponentially with the number of training data and that the dataset we trained is preprocessed by clustering (indicates the data should be similar to each other to some degree), we believe we may acquire results with similar quality if we just train some instead of all data in a cluster. Therefore, we adapt a random sample concept to randomly select a certain percentage of data in the clusters, and feed those random samples into the Ranking-SVM. Table 5.1 shows the total execution time required for training Ranking-SVM in all 48 clusters. (Since we have 48 clusters, we need to train 48 different Ranking-SVM.) The first column indicates the percentage of the whole dataset in clusters trained by the support vector machine. The second and third column denotes the required time in the unit of seconds

and days. Graph one is interpreted from table one. Figure 5.3 is the graphical interpretation of Table 5.1.

Table 5.1 Execution time required for training Ranking-SVM with different percentage of whole dataset in clusters

Training Percentage	Execution Time (Sec)	Execution Time (Days)
100%	811631	9.4
90%	407607	4.7
80%	279522	3.2
70%	163686	1.9
60%	79451	0.9
50%	37551	0.4
40%	13796	0.2

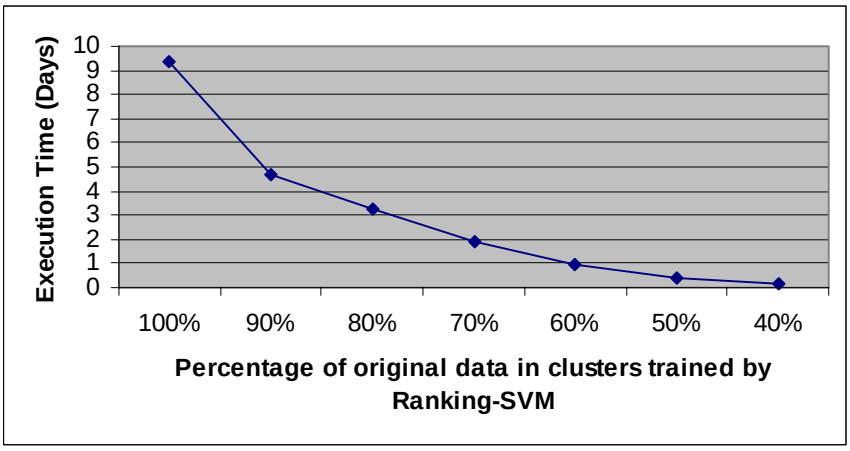


Figure 5.3 Execution time in the unit of days required for training Ranking-SVM with different percentage of whole dataset in clusters.

It can be easily interpreted from figure 5.3 that we can save execution time dramatically by training partial data in each cluster; but the question is, how is the quality of our ranking support vector machine if we only give partial data for training? The next section gives the answer to this question.

5.4.2 Quality Comparison

In Table 5.2 and 5.3, the number of clusters which contain higher than 60% and 70% structural similarity generated by different methods is given. The first column denotes the percentage of original data filtered in each cluster; the first row indicates what percentage of the whole dataset in each cluster trained by Ranking-SVM, and the last position indicates eliminated features by the shrink approach. Thus, the number "35" in row7 (30%) and column4 (80%) of table2 means "If we filter out 30% of the data segments by using Ranking-SVM trained from 80% of the whole dataset in each cluster, we may get 35 clusters with secondary structure greater than 60%."

Table 5.2 Number of clusters with secondary structure similarity > 60% comparison

Train / Filter	all	90%	80%	70%	60%	50%	40%	Shrink
0%	12	12	12	12	12	12	12	12
10%	21	20	22	18	18	17	17	15
15%	24	25	23	23	22	20	21	15
20%	28	27	29	25	28	24	22	14
25%	31	31	30	30	29	28	25	15
30%	36	35	35	32	31	30	30	17
35%	40	39	40	37	36	34	33	18
40%	45	43	42	41	39	37	37	19

Table5.3 Number of clusters with secondary structure similarity > 70% comparison

Train / Filter	all	90%	80%	70%	60%	50%	40%	Shrink
0%	0	0	0	0	0	0	0	0
10%	1	0	2	1	1	0	1	0
15%	3	2	2	1	2	2	1	0
20%	6	4	4	2	2	4	1	0
25%	7	7	6	6	4	3	3	1
30%	9	8	8	8	8	5	5	1
35%	12	10	10	9	8	7	7	1
40%	16	13	15	13	11	11	10	1

Table5.4 Comparison of HSSP-BLOSUM62 2.0 measure

Train Filter	all	90%	80%	70%	60%	50%	40%	Shrink
0%	.749	.749	.749	.749	.749	.749	.749	.749
10%	1.05	.918	.814	.660	.663	.800	.791	.634
15%	.828	.733	.745	.799	.698	.742	.696	.731
20%	.751	.888	.881	.786	.832	.772	.705	.818
25%	.852	.853	.796	.879	.737	.750	.838	.760
30%	.910	.819	.868	.803	.743	.772	.779	.655
35%	.793	.773	.765	.815	.813	.841	.844	.897
40%	.658	.729	.692	.593	.694	.736	.696	.694

Table 5.5 Comparison of DBI Measure

Train Filter	all	90%	80%	70%	60%	50%	40%	Shrink
0%	6.39	6.39	6.39	6.39	6.39	6.39	6.39	6.39
10%	6.23	6.27	6.25	6.19	6.37	6.22	6.17	6.14
15%	6.21	6.09	6.04	6.21	6.22	6.10	6.05	6.03
20%	6.04	6.10	5.96	6.00	6.11	6.02	6.05	5.90
25%	5.98	6.01	5.88	5.84	5.97	5.96	5.90	5.79
30%	5.91	5.91	5.83	5.84	5.79	5.80	5.81	5.79
35%	5.84	5.79	5.82	5.81	5.77	5.77	5.76	5.75
40%	5.70	5.74	5.66	5.67	5.70	5.76	5.68	5.64

To the best of our knowledge, there are no related works using Ranking-SVM on the data of protein sequence motif information transcending protein family boundaries for feature elimination. Therefore, we use the shrink approach which is a traditional clustering improvement for comparison. The average HSSP-BLOSUM62 2.0 value on high structural similarity (>60%) clusters and the DBI measures are available in table4 and 5.

The results of Table 5.2 through Table 5.5 reveal that the quality of clusters improved in all three measures steadily by filtering out part of the original data. Compared to the Shrink approach from a secondary structure similarity point of view, it is not hard to tell that ranking

SVM generates much better results no matter what percentage of the whole dataset are trained. The support vector machine approach produces more clusters with higher than 60% structural similarity almost all of the time. If we compare the number of clusters that share over 70% structural similarity, Ranking SVM unquestionably surpasses the shrink approach. It indicates that our proposed model has a high potential for bringing forth high quality protein sequence motif information.

In our HSSP-BLOSUM62 2.0 measurement, the ranking support vector machine gives a higher value almost all of the time. It implies that our model generates more bio-chemically meaningful motif information by ruling out some less meaningful data points. When it comes to DBI measurement, which is a purely computer scientific aspect evaluation, shrink method always receives lower (indicates better) value. This is mainly because the shrink method is based on simply narrowing the cluster size from outside; in other words, it focuses on shorter intra-cluster distance. Therefore, it can always generate the best DBI value. Although the SVM approach has larger DBI values, the difference between methods is small. More importantly, the ranking SVM shows the same tendency of decreasing DBI measure as with the shrink approach.

If we consider Ranking-SVM trained with all data segments in each cluster and try to find the optimal filtering percentage based on the results shown above filtering 30% of the whole data size seems to yield the best results. The motif information we want transcends protein family boundaries; thus, if we filter out too many segments, we might generate motif information falling into some specific protein family. The reason we choose 30% as representative is that it matches the criteria of filtering part of the data and creates higher structural similarity results. Considering the evaluation of secondary structural similarity greater than 60%, it is the first big improvement (an extra five clusters) compared with filtering out 25%. More importantly, it

achieves very high HSSP-BLOSUM62 2.0 values, which indicates the motif information is bio-chemically meaningful.

Compared with Ranking-SVM trained by different percentages of the original data, we can tell that the more training data, the better quality generated. However, the results generated from training all data, 90%, and 80% are very similar to each other in all evaluation measures. The quality starts to go down slightly when Ranking-SVM is trained with less than 70% of data in each cluster. Figure 5.4, 5.5 and 5.6 are derived from Table 5.2 to 5.5 when 30% of the original dataset are filtered. The results support our assumption that we may acquire very similar results if we just train partial, instead of all, data in a cluster; and according our figures and tables, 80% seems to be the best number for this "partial" value. It requires only 1/3 execution of training all data, shows a similar number of clusters with high structural similarity, yields high HSSP-BLOSUM62 2.0 value, and reduces the DBI measure.

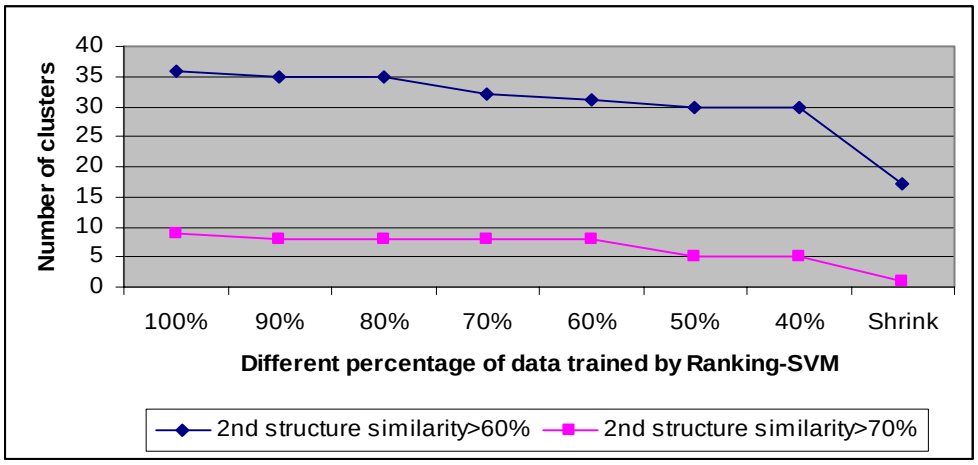


Figure 5.4 Comparison of number of clusters with secondary structure similarity greater than 60% and 70% when different percentage of data in each cluster trained by Ranking-SVM, when 30% of original data been filtered.

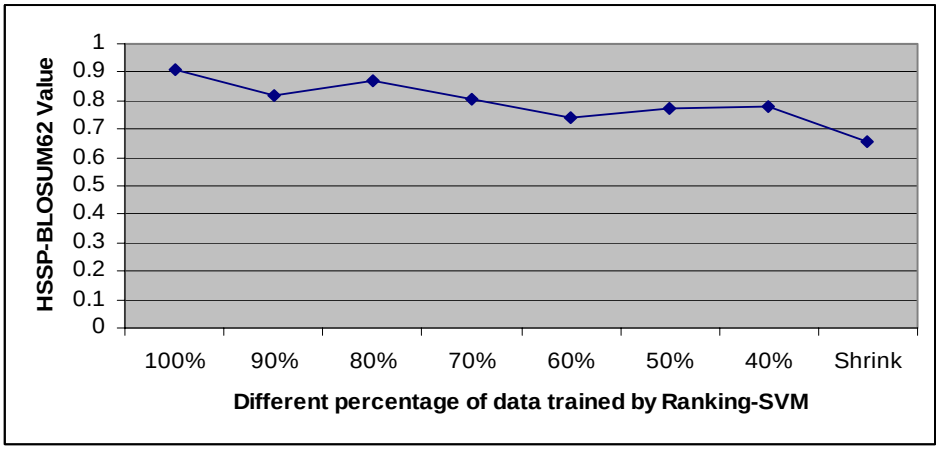


Figure 5.5 Comparison of HSSP-BLOSUM62 2.0 value when different percentage of data in each cluster trained by Ranking-SVM, when 30% of original data been filtered.

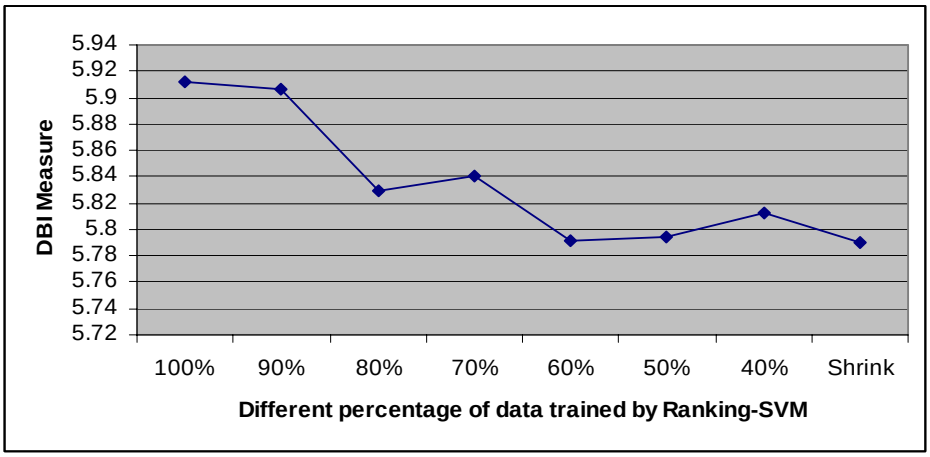


Figure 5.6 Comparison of DBI measure when different percentage of data in each cluster trained by the Ranking-SVM, when 30% of original data been filtered.
(Lower indicates better)

5.4.3 Sequence Motifs

We select some representative motif information generated from filtering 30% of the whole dataset by using Ranking-SVM trained with 80% of data in each cluster. Figure 5.7 to 5.11 illustrates five different sequence motifs before and after feature elimination. In this chapter, instead of using the existing format [8, 36, 37], we propose a new representation format

combined with amino acid logo [38] to show the motif we discovered from our new approach. By using this new format, the frequency of each amino acid can be easily interpreted; and more importantly, this is what biologists use also. We believe that this new format will be adapted and used in related research of finding sequence motifs that transcend protein family boundaries.

- The upper box gives the number of members belonging to this motif, the secondary structural similarity and the average HSSP-BLOSUM62 2.0 value.
- The graph demonstrates the type of amino acids which frequently appeared in the given position by amino acid logo. It only shows the amino acids appearing with a frequency higher than 8%. The height of symbols within the stack indicates the relative frequency of each amino or nucleic acid at that position.
- The x-axis label indicates the representative secondary structure to the position.

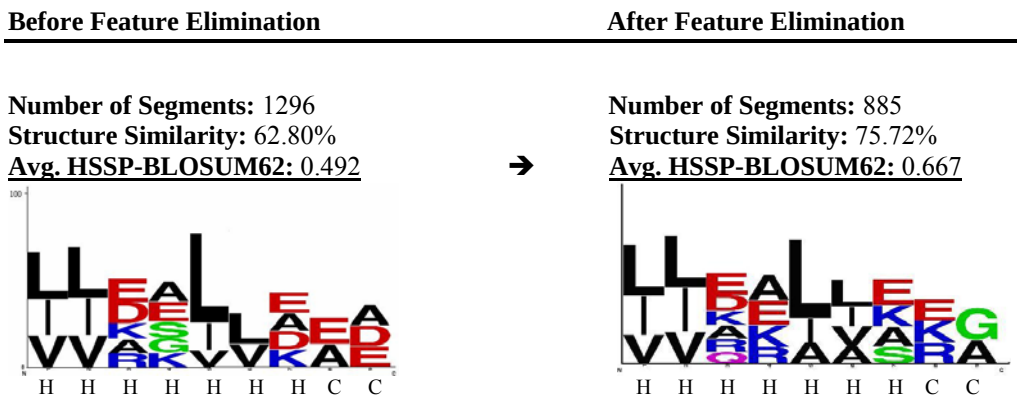


Figure 5.7 Helix-Coil motif

Before Feature Elimination		After Feature Elimination
Number of Segments: 867 **Structure Similarity:** 52.62% **Avg. HSSP-BLOSUM62:** 1.232	➔	**Number of Segments:** 730 **Structure Similarity:** 65.51% **Avg. HSSP-BLOSUM62:** 0.890

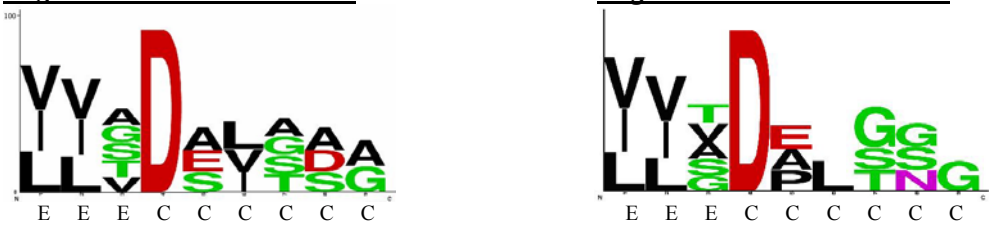


Figure 5.8 Sheet-Coil motif

Before Feature Elimination		After Feature Elimination
Number of Segments: 1556 **Structure Similarity:** 59.25% **Avg. HSSP-BLOSUM62:** 0.913	➔	**Number of Segments:** 1197 **Structure Similarity:** 73.35% **Avg. HSSP-BLOSUM62:** 0.697

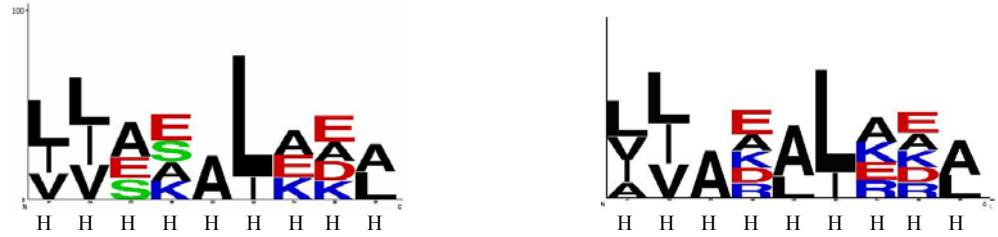


Figure 5.9 Helix motif

Before Feature Elimination		After Feature Elimination
Number of Segments: 520 **Structure Similarity:** 65.66% **Avg. HSSP-BLOSUM62:** -0.345	➔	**Number of Segments:** 360 **Structure Similarity:** 74.68% **Avg. HSSP-BLOSUM62:** 1.284

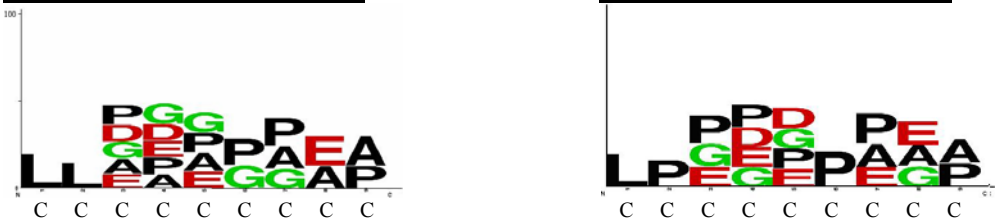


Figure 5.10 Coil motif

Before Feature Elimination		After Feature Elimination
Number of Segments: 177 **Structure Similarity:** 51.09% **Avg. HSSP-BLOSUM62:** 0.296	➔	**Number of Segments:** 122 **Structure Similarity:** 61.98% **Avg. HSSP-BLOSUM62:** 0.435

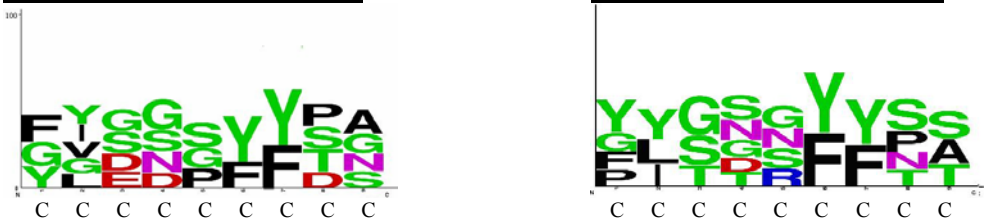


Figure 5.11 Coil motif

After the experiment, we applied our feature elimination model on all of our FGK-250's 799 clusters. Table 5.6 gives the details of the improvements on each information granule. The first row shows the information granule number; the second row gives the total number of clusters in each information granule. The fourth row presents the number of clusters with secondary structure similarity between 60% and 70% before the feature elimination process. The fifth and sixth rows follow the trend. The eighth row gives the information about the number of clusters with secondary structure similarity between 60% and 70% after the feature elimination process. The last two rows follow the same trend as the eighth row. For example, if we take a look at the fifth column, the fifth row and the ninth row, the information can be interpreted as "For information granule three, before the feature elimination process, it has 4 clusters with secondary structural similarity between 70% and 80%. After the process, the number of clusters with secondary structural similarity between 70% and 80% increases to 16." According to Table 5.6, it is easy to tell that the quality of the clusters improves dramatically. The total number of clusters with secondary structural similarity higher than 60% increased from 343 to 543; and the number of clusters with secondary structural similarity higher than 70% increased from 112 to

256. We believe this extracted motif information can be the key to discovering the relation between protein primary structure and tertiary structure.

Table 5.6 Number of high quality motifs in each information granule before and after the Super GSVM-FE training on 80% of the cluster members

	G0	G1	G2	G3	G4	G5	G6	G7	G8	G9	ALL
Total number of clusters	151	76	95	72	70	133	143	5	48	6	799
Original FGK-250											
60%~70%	36	24	24	28	32	31	35	2	15	4	231
70%~80%	21	3	12	4	4	24	20	0	0	0	88
>80%	7	0	7	0	0	4	6	0	0	0	24
After Super GSVM-FE											
60%~70%	44	30	31	30	42	39	40	3	24	4	287
70%~80%	27	17	17	16	16	27	30	1	4	1	156
>80%	26	2	19	2	1	26	23	0	0	1	100

5.5 Analysis of the Extracted Rules

To further analyze the extracted 543 motifs, we perform local secondary structure prediction based the original and extracted information.

The latest release of PISCES includes 4345 PDB files. Compare with the dataset in our experiment, 2419 PDB files are excluded. Therefore, we regard our 2710 protein files as the training dataset and 2419 protein files as the testing dataset. We convert the testing dataset by the approach we introduced in section 2.2; more than 520,000 segments are generated. The dataset contains 38.44% of Helixes, 23.37% of Sheets, and 38.19% of Coils. The prediction procedure is straightforward: we calculate the distance between the testing segment and motif information; if the distance is within the threshold, we predict the testing segment has the same

secondary structure as the motif. During the process, no decision fusing or voting scheme is involved.

Table 5.7 Comparison of prediction results on the original motif information and the extracted motif information

		Original Rules			Extracted Rules		
Distance Threshold	Motif Quality	#motifs Participate	#segments Predicted	Accuracy	#motifs Participate	#segments Predicted	Accuracy
500	>80%	22	116	93.82%	79	361	90.94%
	70%~80%	67	304	86.99%	81	446	85.02%
	60%~70%	114	1608	75.95%	102	1791	68.52%
600	>80%	24	2570	87.10%	100	7522	87.64%
	70%~80%	88	6725	80.11%	155	9358	75.96%
	60%~70%	230	26162	66.57%	278	31783	65.71%
700	>80%	24	20997	76.70%	100	67310	78.22%
	70%~80%	88	66096	69.03%	156	94603	64.42%
	60%~70%	231	214652	55.20%	287	258488	54.92%
800	>80%	24	106445	63.90%	100	365572	65.71%
	70%~80%	88	364096	57.86%	156	554660	53.33%
	60%~70%	231	1069434	47.98%	287	1274296	48.07%

We compare the results with different distance thresholds generated from original motif information and extracted motif information. Due to the fact that higher quality motif information has better prediction strength, we group our motifs into three categories: motifs with 2^{nd} structure similarity higher than 80%, from 70% to 80%, and from 60% to 70%. Table 5.7 is the prediction results for different groups. The first column shows the distance threshold we set for the prediction experiment. The second column indicates three different groups of motif quality. The third and sixth column give the number of motifs involved with the prediction process. (If the distance threshold is strict, some motifs may not find similar data segments.) The fourth and seventh illustrate the total number of segments that has been predicted by the group of motifs. The fifth and the last column pertain to the prediction accuracy. For each data

segment, due to the window size setup, nine positions of the 2nd structure are predicted. The accuracy is calculated by the number of correctly predicted positions divided by the total number of predicted positions. One important thing to notice is that it is a three prediction (helix, sheet or coil) instead of bi-class prediction. Therefore, it is harder than the traditional yes or no problem.

By interpreting the Table 5.7, it is easy to realize that if we set a stricter distance threshold, the numbers of motif participate and the numbers of predicted segments decrease, and the prediction accuracy increase. Distance threshold equals 500 is a typical restrict example, both the original and the extracted motifs can only identify around 2000 segments and yield very precise 2nd structure prediction. The experiment with distance threshold 600 shows the prediction accuracy is almost equal to the quality of the motifs. Most of the motifs can identify some specific segments. In this experiment, the extracted motifs cover more data segments, while the prediction accuracy is very compatible with the original motifs. When the distance threshold is loose, it appears to predict many more data segments. However, the quality of the results drops dramatically. The experiment with 800 distance threshold seems a good example of this.

We also calculate the overall prediction accuracy in terms of using motifs information with structural similarity higher than 80%, 75%, 70%, 65%, and 60% on distance threshold equals 600 and 70. Table 5.8 gives the comparison table and Figure 5.12, 5.13 are interpreted from the table. The major difference between Table 5.7 and 5.8 is on the second column; in table 5.8, ">70%" indicates that the motifs structural quality higher than 70%, which includes the group of motifs with 2nd structure similarity higher than 80% and among 70% to 80%, where ">80%, >75%, >65%, >60%" follow the same trend.

Table 5.8 Comparison of overall prediction results on the original motif information and the extracted motif information

		Original Rules			Extracted Rules		
Distance Threshold	Motif Quality	#motifs Participate	#segments Predicted	Accuracy	#motifs Participate	#segments Predicted	Accuracy
600	>80%	24	2570	87.10%	100	7522	87.64%
	>75%	66	6062	85.56%	166	10873	84.95%
	>70%	112	9295	82.30%	254	16880	80.95%
	>65%	192	16746	76.67%	398	32707	74.63%
	>60%	342	35457	70.48%	532	48663	70.94%
700	>80%	24	20997	76.70%	100	67310	78.22%
	>75%	66	53580	75.08%	167	104694	74.78%
	>70%	112	87093	70.99%	256	161913	70.02%
	>65%	193	152106	65.17%	401	288086	64.11%
	>60%	343	301745	59.64%	543	420401	60.70%

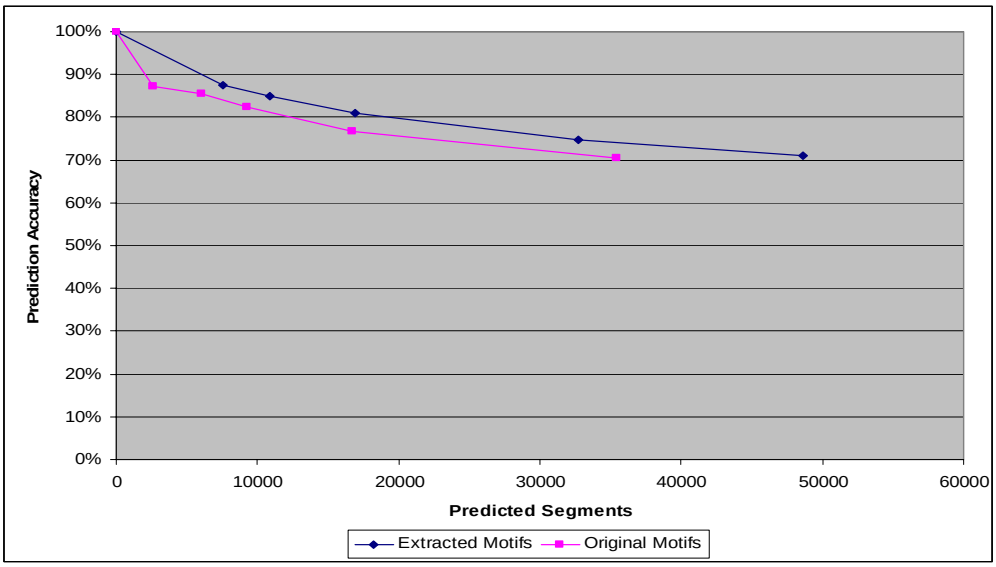


Figure 5.12 The relation between prediction accuracy and the number of predicted segments when distance threshold equals 600

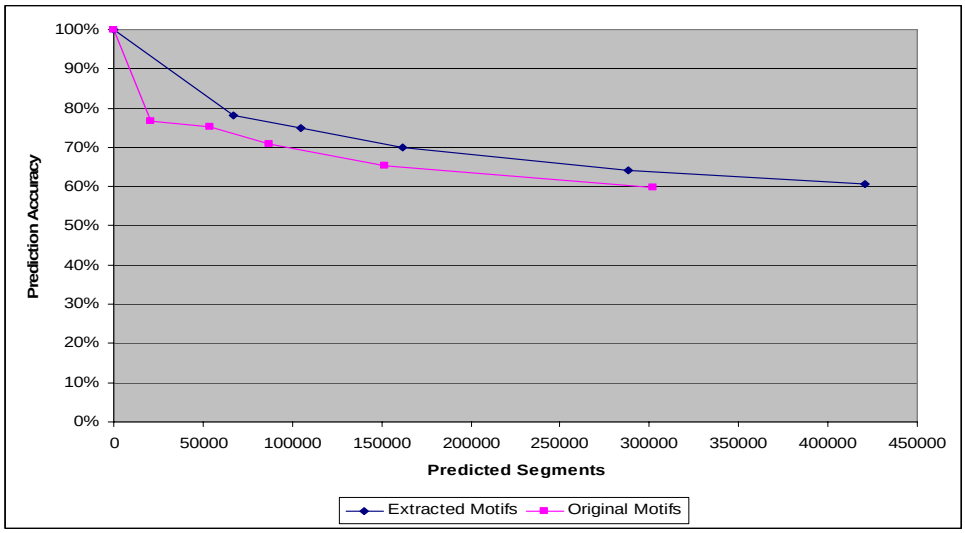


Figure 5.13 The relation between prediction accuracy and the number of predicted segments when distance threshold equals 700

According to the information given in Figure 5.12 and 5.13, we realize that the extracted motifs show better predicted segments coverage than the original motifs as well as the prediction accuracy. In this section, our experiment shows that the motifs generated from Super GSVM-FE model are meaningful. And more importantly, it suggests that the motifs we found are not sequence motifs only, but also structure motifs.

5.6 Conclusions

A novel granular feature elimination model called Super GSVM-FE which combines Fuzzy C-means, Greedy K-means clustering algorithm and Ranking SVM has been proposed to extract protein sequence motif information. In this model, we utilize fuzzy clustering to split the whole dataset into several information granules and analyze each granule by Greedy K-means

clustering algorithm. After that, we rate all members in all clusters by ranking SVM, and then filter out less meaningful segments to obtain higher quality motif knowledge. Analysis of sequence motifs also shows that filtering some portion of the original dataset may reveal some subtle motif information hidden by noisy data points. This is the first time that we justify the need for feature elimination, in the dataset of protein sequence motif transcending protein family boundaries, by providing two major reasons: 1. the information we try to generate is about sequence motifs, but the original input data are derived from whole protein sequences by the sliding window technique. 2. During fuzzy c-means clustering, it has the ability to assign one segment to more than one information granule. However, not all data segments have direct relation to the granule assigned.

Additionally, we performed a comprehensive analysis on the tradeoff between the execution time and the quality of motif information. Since we performed Ranking-SVM on clusters, trained data are similar to each other to some degree. Therefore, based on our results, training 80% of the original data can not only save training time but also obtain a competitive quality of extracted protein sequence motif information. It opens a new research direction with the cluster support vector machine. We believe some other research with a huge input data size may adapt our model to generate high quality filtered results.

CHAPTER 6

SUPER-RULE-TREE (SRT) STRUCTURE CONSTRUCT BY NOVEL HYBRID HIERARCHICAL K-MEANS (HHK) CLUSTER ALGORITHM

Protein sequence motifs are gathering more and more attention in the field of sequence analysis. These recurring patterns have the potential to determine a protein's conformation, function, and activities. In order to identify these motifs, most of the enumerative algorithms need to specify the size of the motif in advance. Because of the fixed size, they often deliver a number of similar motifs *(1)including mismatches* or *(2)shifted by one base* [39], which is problematic. The first problem implies that some group motifs may be similar to one another. The second problem probably can be more easily seen in this way: If there exists a biological sequence motif with length of 12 and we set the window size to 9, it is highly possible that we discovered two similar sequence motifs where one motif covers the front part of the biological sequence motif and the other one covers the rear part. In this chapter, we deal with the first problem and manage the second one in next chapter.

Dealing with the first problem would probably be easier than the second one, since we may use the Super-rules concept [40] to cluster those motifs and find the similarities among them. Two of the most popular algorithms for distance-based clustering are Hierarchical clustering [41] and K-means clustering [42]. According to Hu et al [43], many improvements to these two famous clustering algorithms have been proposed [44-48]; however, they adapt our Hybrid Hierarchical-K-means (HHK) clustering algorithm [49], which directly combines the two classic methods and

yield good results. In this chapter, we proposed the latest version of HHK, which requires no predefined parameters to construct a Super-Rule-Tree structure.

6.1 Novel Hybrid Hierarchical K-means (HHK) Clustering Algorithm

Clustering is a technique to divide datasets into several subsets whose elements share similar attributes. Among clustering algorithms, Hierarchical and K-means clustering are the two most popular and classic methods. However, both have their innate disadvantages. Hierarchical clustering cannot represent distinct clusters with similar expression patterns. Also, as clusters grow in size, the actual expression patterns become less relevant. K-means clustering requires a specified number of clusters in advance and chooses initial centroids randomly; in addition, it is sensitive to outliers.

We present a hybrid approach to combine the merits of the two classic approaches and discard disadvantages we mentioned. A brief description of HHK clustering algorithm follows. First, we carried out agglomerative hierarchical clustering and let the program stop at a certain terminal point (a user defined percentage which is determined by the whole clustering process carried out by hierarchical clustering). From the clusters generated from hierarchical clustering, we computed the mean value of each cluster as the initial point for k-means to obtain the initial centroid. Also, the number of clusters generated from hierarchical clustering is k-mean's number of clusters. After that, we worked on k-means clustering with which every cluster MUST at least contain the same objects generated from hierarchical clustering. This is due to the fact that hierarchical clustering had already put objects that were very close with one another into clusters, and the goal of k-means clustering is to put close objects together, which is in the same

direction as what hierarchical clustering accomplished. Therefore, we can trust the results of hierarchical clustering.

We apply HHK clustering algorithm for super-rules [40] generation in this chapter. In order to avoid human intervention and let the Super-rule present the original data nature, we modified our HHK clustering algorithm to become a fully parameter-free algorithm. The original HHK required q user to decide when to stop the hierarchical clustering and proceed to K-means clustering. Since the role of HHK clustering algorithm is to generate the super-rules, the results of the clustering should be as detailed as possible. Therefore, the approach we propose to avoid the parameter setup is to let the agglomerative hierarchical clustering complete execution, and we record the number of clusters it generated. After that, we carry out the HHK clustering algorithm and let the hierarchical clustering stop when it generates the largest number of clusters. The reason for this process is that while the hierarchical clustering stops at the point we mentioned, the HHK clustering may generate the largest number of super-rules as well as the most detailed information. We may apply the HHK on the super-rules again to generate super-super-rules if necessary. By this manner, we can form a Super-Rules-Tree (SRT) structure. The HHK clustering is summarized in Figure 6.1

(1) Finish a complete agglomerative Hierarchical clustering on the data and record the number of clusters generated during the process.
(2) Run the agglomerative Hierarchical clustering again and stop the process when the largest number of clusters is generated.
(3) Execute the K-means clustering on the remaining data which are not processed in step (2) use the centroids for every cluster generated in step (2) are served as the initial centroids in the K-means clustering algorithm.

Figure 6.1 The HHK Clustering Algorithm

6.2 Super-Rule-Tree (SRT) Structure

In Zhong's work [8], 253 sequence motifs with high structural similarities are revealed by their improved K-means clustering algorithm with the fixed window size 10, and those motifs are grouped into 27 major patterns according to their common characteristics. This suggests that many motifs are similar to one another. Since the dataset we used is very similar to [8], we both selected from PISCES [16] (our PISCES list was more updated) and expended by HSSP , we believe that our results which come from our Fuzzy Greedy K-means (FGK) model [37] should have a similar trend. Therefore, we perform HHK clustering algorithm on our 343 motifs for Super-Rule-Tree generation. As we discussed in section 2A, we carry out a complete hierarchical clustering and record the number of clusters generated during the process as shown in Figure 6.2.

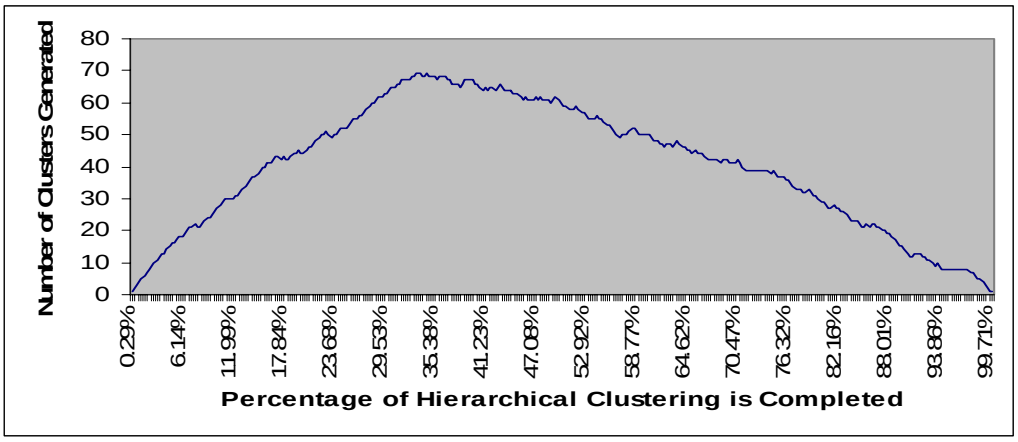


Figure 6.2 The relation between percentages of Hierarchical clustering is completed and the numbers of clusters are generated for level-1 super-rule generation.

It is clear that a peak is found during 33.63% of the clustering and it generated 69 clusters. After we obtain this information, we may start the HHK clustering: initially, we run the

hierarchical clustering until 69 clusters are generated; after that, by using the center of these 69 clusters as the initial centroids, we run K-means clustering algorithm for the remaining motifs (the motifs that have not been clustered yet). After 69 level-1 super-rules are generated, since the number of super-rules is still large, we perform another HHK clustering. Figure 6.3 is the analysis of when to stop the Hierarchical clustering.

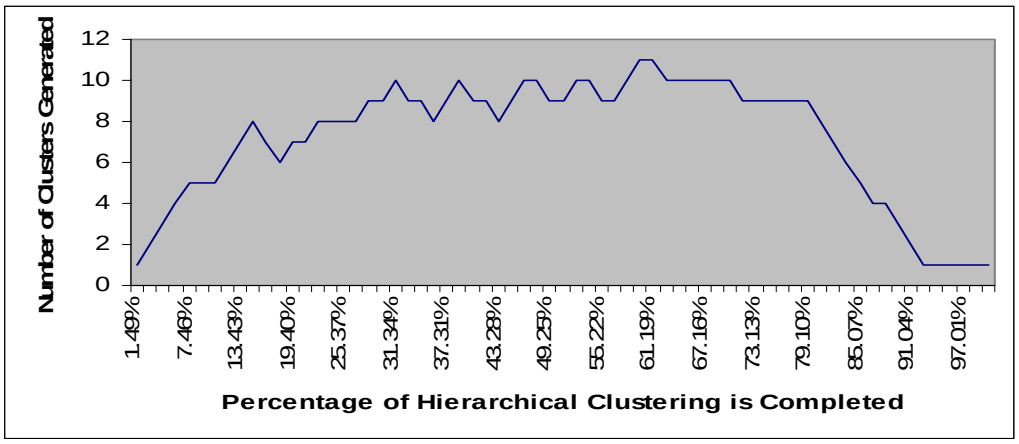


Figure 6.3 The relation between percentages of Hierarchical clustering is completed and the numbers of clusters are generated for level-2 super-rule generation.

After 69 level-1 super-rules are generated, since the number of super-rules is still large, we perform another HHK clustering on it. Figure 6.3 is the analysis of when to stop the Hierarchical clustering. Based on Figure 6.3, we run Hierarchical clustering algorithm for 61.19% and it generates 11 clusters. In the end, we construct a SRT as shown in Figure 6.4.

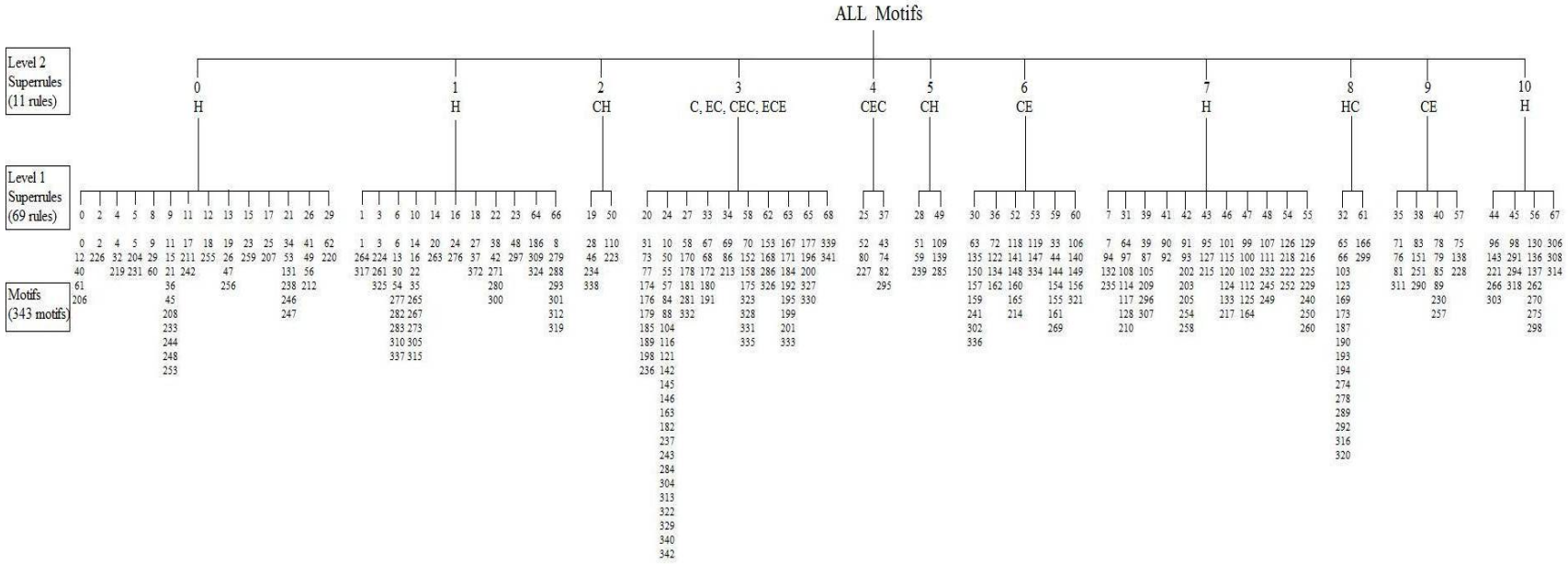


Fig. 6.4 The Super-Rule-Tree (SRT) of 343 different sequence motifs

By further analysis of the Super-Rule-Tree, we made note that the result of level 1 super-rule is grouping motifs with similar sequences (protein primary structure) and we also made note that all groups share common secondary structures; the outcome of level 2 super-rule is more likelyto put motifs with similar secondary structure into groups. We mark representative 2nd structure on the level 2 super-rules in the Figure 6: Helix (H), Coil (C), Coil-Helix (CH), Helix-Coil (HC), Coil-Sheet (CE), Sheet-Coil (EC), Coil-Sheet-Coil (CEC), and Sheet-Coil-Sheet (ECE). More specifically, a representative secondary structure identified as Helix (H) when all of a motif's secondary structures are Helix (same logic applies to C), and a complex folding like "CH" means that the secondary structures of a motif are initially composed of Coil and then turn to Helix. Considering level 2 super-rules, all have a very consistent representative 2nd structure except super-rule number three, which is a group of mixing Coils and Sheets. These results suggest that the Super-Rule-Tree (SRT) gives a good overlook of the large amount of rules (motifs); people can easily recognize the similarity among rules and rules. By looking at Figure 6, we may notice that the majority are the Helix motifs. Because the statistical analysis of the structural database indicates the average length of Helices is ten [8] and the window size we set in our previous work is nine, 70% of the sequence motifs generated by our FGK model are related to Helices.

Figure 6.5 gives an example of level 1 super-rule 28, which belongs to level 2 super-rule 5 (CE), and its components: motif #51, 59 and 239. The motif presentation format is combined with amino acid logo [38]:

- The upper box gives the motif ID number, the number of members belonging to this motif, and the average secondary structural similarity.

- The graph demonstrates the type of amino acid frequently appearing in the given position by amino acid logo. It only shows the amino acid appearing with a frequency higher than 8%. The height of symbols within the stack indicates the relative frequency of each amino or nucleic acid at that position.
- The x-axis label indicates the representative secondary structure and the structural similarity among all members to the position. For example, H70 indicates the representative 2nd structure is Helix and 70% of the whole members' 2nd structure is Helix to this position.

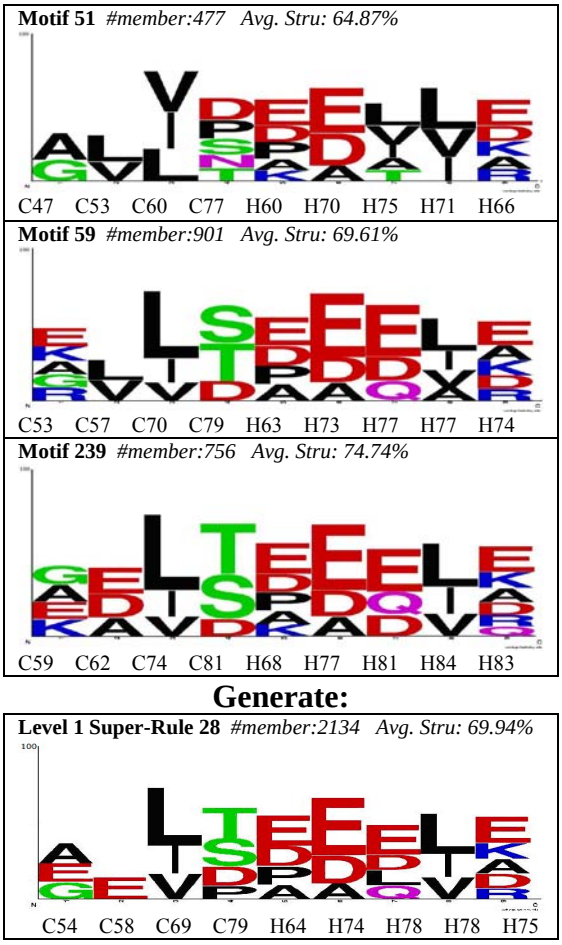


Fig. 6.5 Example of level 1 super-rule #28 generated from motif #51, 59, 239

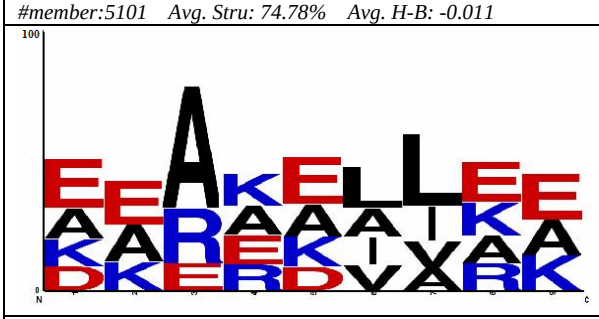
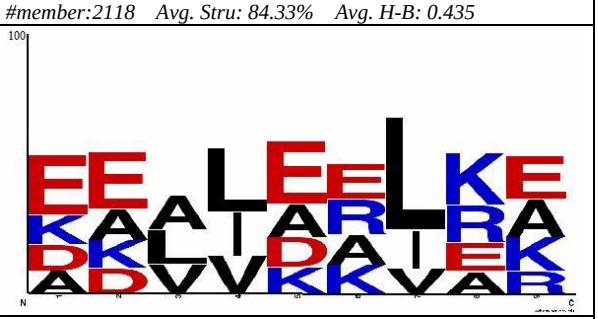
By analyzing Figure 6.5, we can recognize that although the clustering is performed on the sequence (primary structure) base only, the secondary structures among these three motifs are almost the same. The motifs #51, 59, and 239 not only all constructed in a Coil-Helix form, but also all start from two weak Coils and then turn into a higher secondary structural similarity to the next two Coils followed by strong five-position Helixes. We may also tell from the Figure 6.5 that the super-rule is an adequate representation of these three motifs.

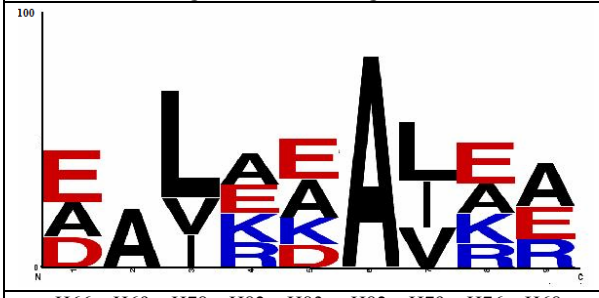
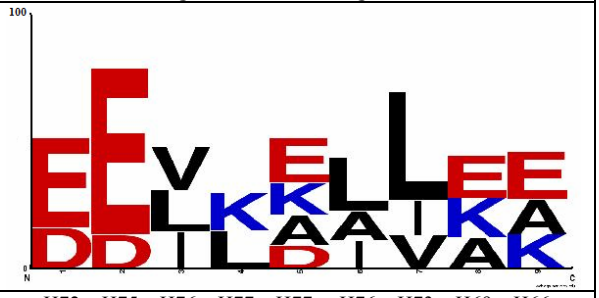
6.3 Level-1 Super-Rule Motifs

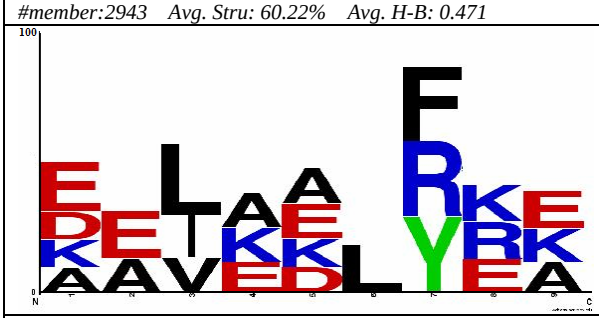
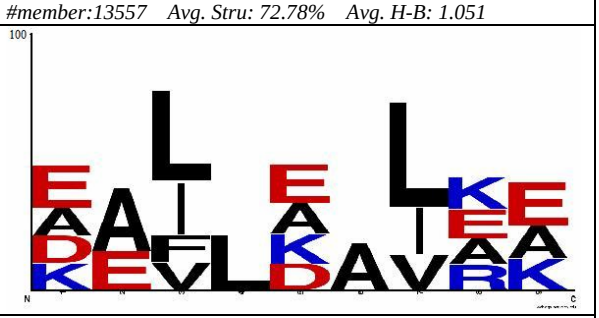
Since we summarized our 343 motifs into 69 level-1 Super-rules, we present these Super-rules in this section by the order of the appearance in level-2 Super-rules. The motif presentation format is described below:

- The upper box gives the level-1 Super-Rule ID number, the number of members belonging to this motif, the average secondary structural similarity and the average HSSP-BLOSUM62 2.1 value.
- The graph demonstrates the type of amino acid frequently appearing in the given position by amino acid logo. It only shows the amino acid appearing with a frequency higher than 8%. The height of symbols within the stack indicates the relative frequency of each amino or nucleic acid at that position.
- The third row indicates the representative secondary structure and the structural similarity among all members to the position.
- The last row shows the representative HSSP-BLOSUM62 2.1 value to the position.

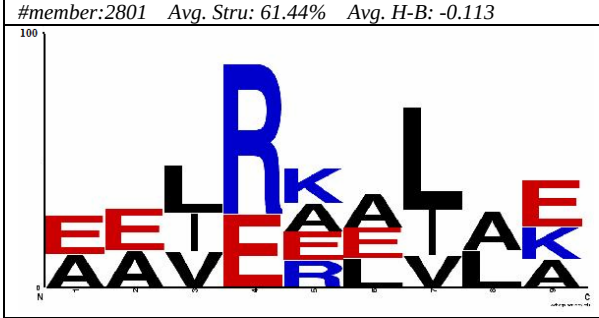
Table 6.1 Level 2 Super-Rule 0 (H motifs)

Level 1 Super-Rule 0 *#member:5101 Avg. Stru: 74.78% Avg. H-B: -0.011*	**Level 1 Super-Rule 2** *#member:2118 Avg. Stru: 84.33% Avg. H-B: 0.435*
	
H73 H76 H78 H79 H79 H77 H74 H71 H76	H79 H82 H85 H90 H89 H89 H87 H82 H76
-0.1 -0.3 -0.8 0.0 -0.2 0.67 0.91 0.09 -0.3	0.08 -0.0 -0.0 1.89 -0.0 0.05 1.75 0.36 -0.1

Level 1 Super-Rule 4 *#member:4104 Avg. Stru: 75.89% Avg. H-B: 1.139*	**Level 1 Super-Rule 5** *#member:3008 Avg. Stru: 73.44% Avg. H-B: 0.581*
	
H66 H69 H78 H82 H83 H82 H79 H76 H68	H72 H75 H76 H77 H77 H76 H73 H69 H66
-0.2 4 1.70 -0.1 -0.3 4 1.92 -0.0 -0.7	2.0 2.0 1.97 -2.0 -0.2 0.18 1.72 -0.2 -0.3

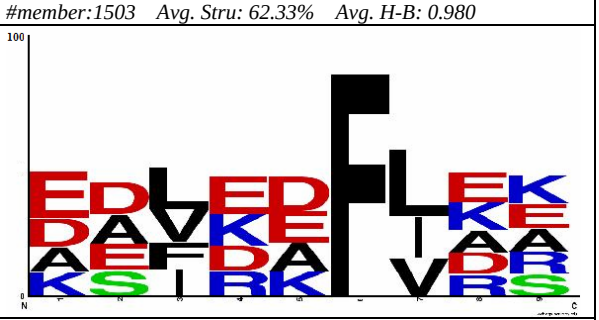
Level 1 Super-Rule 8 *#member:2943 Avg. Stru: 60.22% Avg. H-B: 0.471*	**Level 1 Super-Rule 9** *#member:13557 Avg. Stru: 72.78% Avg. H-B: 1.051*
	
H59 H60 H62 H64 H65 H64 H60 H56 H52	H73 H75 H78 H79 H78 H76 H71 H65 H60
0.02 -1.0 1.84 -0.4 -0.3 4 -0.7 1.02 -0.3	-0.1 -1.0 1.13 4 -0.2 4 1.81 0.08 -0.2

Level 1 Super-Rule 11
#member:2801 Avg. Stru: 61.44% Avg. H-B: -0.113



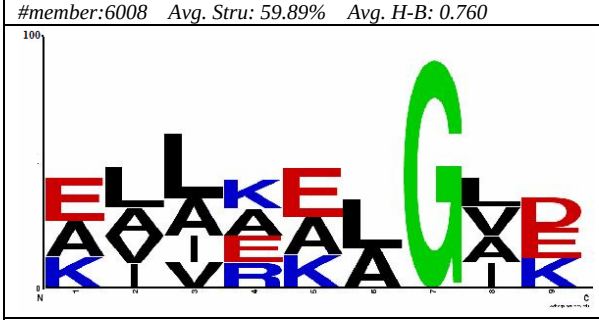
H57	H60	H62	H64	H65	H64	H62	H60	H59
-1.0	-1.0	1.94	0.0	0.07	-1.6	1.79	-1.0	-0.2

Level 1 Super-Rule 12
#member:1503 Avg. Stru: 62.33% Avg. H-B: 0.980



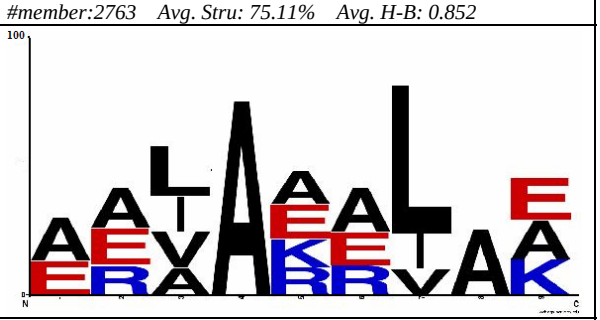
H55	H54	H63	H66	H68	H68	H65	H63	H59
-0.0	-0.0	0.90	0.51	-0.3	6	1.85	-0.2	0.03

Level 1 Super-Rule 13
#member:6008 Avg. Stru: 59.89% Avg. H-B: 0.760



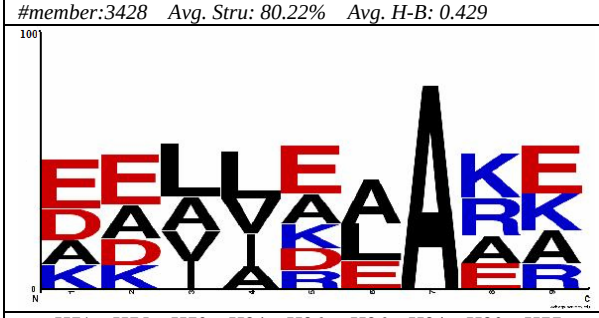
H61	H61	H61	H59	H55	H45	H77	H72	H48
-0.3	0.62	0.44	0.02	-0.3	-1.0	6	0.70	0.68

Level 1 Super-Rule 15
#member:2763 Avg. Stru: 75.11% Avg. H-B: 0.852



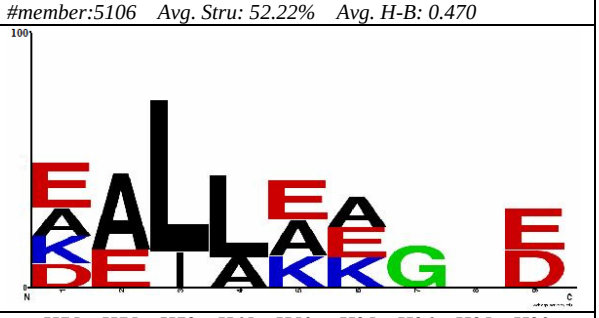
H74	H78	H79	H81	H80	H79	H75	H68	H62
-1.0	-0.7	0.85	4	-0.0	-0.7	1.70	4	-0.3

Level 1 Super-Rule 17
#member:3428 Avg. Stru: 80.22% Avg. H-B: 0.429

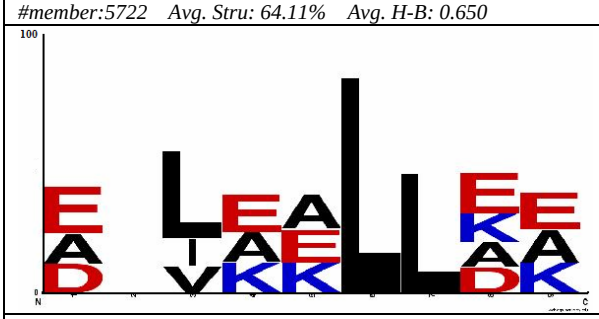


H71	H75	H79	H84	H86	H86	H84	H80	H77
0.00	-0.1	0.61	0.85	-0.2	-1.5	4	0.22	-0.0

Level 1 Super-Rule 21
#member:5106 Avg. Stru: 52.22% Avg. H-B: 0.470



H75	H75	H72	H65	H50	H35	H36	H35	H30
-0.1	-0.1	2.0	-1.0	-0.3	-0.4	3	0	2.0

Level 1 Super-Rule 26 *#member:5722 Avg. Stru: 64.11% Avg. H-B: 0.650*								
								
H59	H61	H67	H71	H70	H69	H64	H61	H55
-0.0	0	1.69	-0.3	-0.4	4	4	-0.1	-0.3

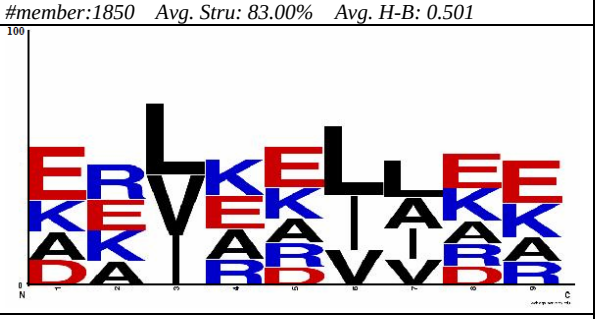
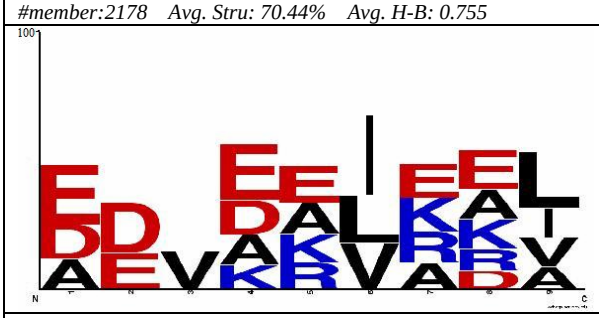
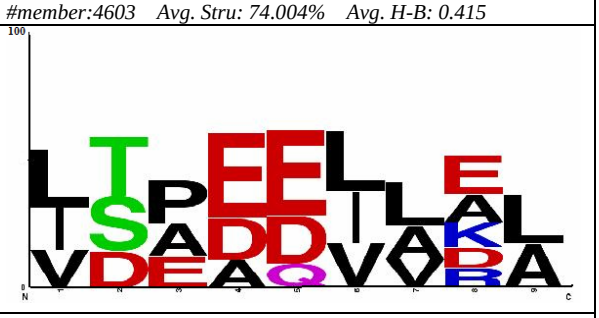
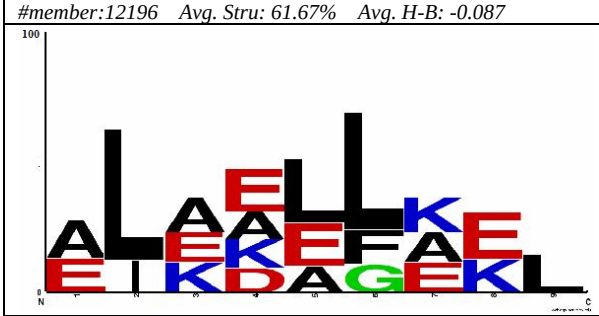
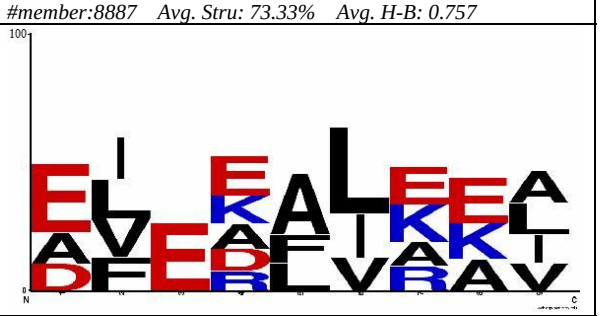
Level 1 Super-Rule 29 *#member:1850 Avg. Stru: 83.00% Avg. H-B: 0.501*								
								
H85	H87	H90	H90	H88	H86	H80	H74	H67
-0.1	0.13	1.90	0.03	-0.1	1.94	0.63	-0.1	0.08

Table 6.2 Level 2 Super-Rule 1 (H motifs)

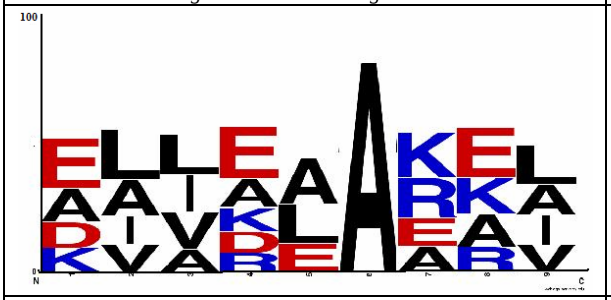
Level 1 Super-Rule 1 *#member:2178 Avg. Stru: 70.44% Avg. H-B: 0.755*								
								
H50	H52	H70	H77	H78	H80	H79	H76	H72
-0.0	2.0	2	-0.0	-0.0	2.16	0.10	-0.2	0.76

Level 1 Super-Rule 3 *#member:4603 Avg. Stru: 74.004% Avg. H-B: 0.415*								
								
C67	C77	H63	H75	H78	H79	H77	H76	H74
1.91	0.16	-1.0	0.28	1.67	1.95	-0.0	-0.2	-1.0

Level 1 Super-Rule 6 *#member:12196 Avg. Stru: 61.67% Avg. H-B: -0.087*								
								
H69	H73	H74	H74	H64	H60	H54	H51	H36
-1.0	2.0	-0.4	-0.1	-2.0	-2.0	-0.30	1.0	2

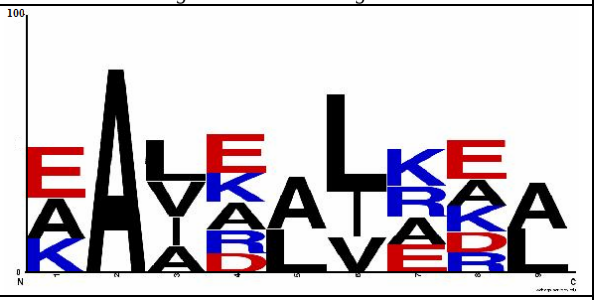
Level 1 Super-Rule 10 *#member:8887 Avg. Stru: 73.33% Avg. H-B: 0.757*								
								
H71	H77	H80	H81	H80	H77	H73	H68	H53
-0.1	0.95	5	-0.1	-1.2	1.82	0.12	-0.3	0.57

Level 1 Super-Rule 14
#member:2644 Avg. Stru: 81.00% Avg. H-B: 0.503



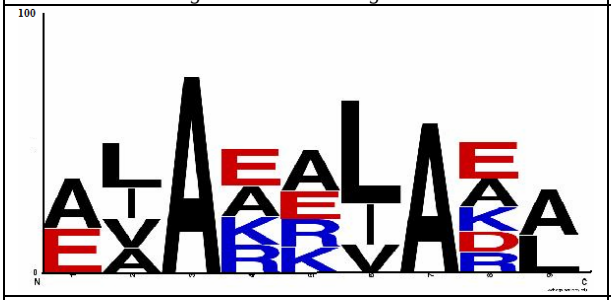
H75	H79	H85	H88	H88	H86	H83	H79	H66
-0.1	0.49	0.97	-0.1	-1.5	4	0.26	-0.0	0.62

Level 1 Super-Rule 16
#member:2927 Avg. Stru: 77.44% Avg. H-B: 0.454



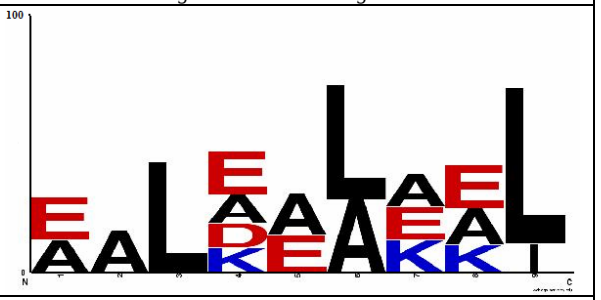
H81	H84	H84	H84	H82	H79	H74	H69	H60
-0.3	4	0.79	-0.1	-1.0	1.8	0.09	-0.1	-1.0

Level 1 Super-Rule 18
#member:3334 Avg. Stru: 73.89% Avg. H-B: 0.896



H71	H74	H78	H80	H79	H78	H74	H70	H61
-1.0	0.79	4	-0.0	-0.2	1.75	4	-0.2	-1.0

Level 1 Super-Rule 22
#member:6540 Avg. Stru: 65.67% Avg. H-B: 0.465

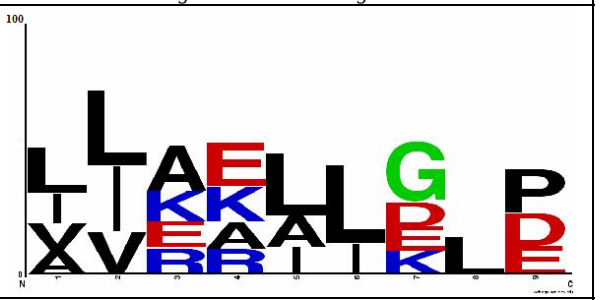


H53	H56	H66	H69	H69	H72	H71	H69	H66
-1.0	2	4	-0.1	-1.0	-1.0	-0.3	-0.3	2.0

Level 1 Super-Rule 23
#member:2583 Avg. Stru: 60.78% Avg. H-B: 0.674

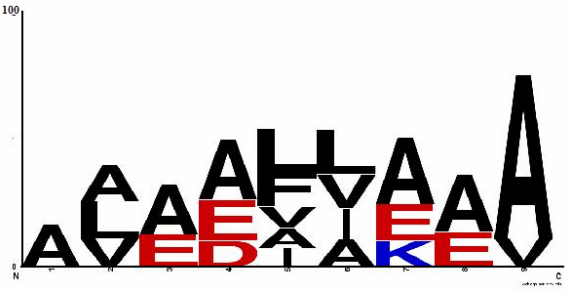
H63	H64	H62	H61	H58	H60	H61	H60	H58
-0.2	2.0	-1.0	-0.2	-1.0	3.0	-0.2	-0.3	4

Level 1 Super-Rule 64
#member:2370 Avg. Stru: 65.44% Avg. H-B: 0.626



H78	H78	H77	H74	H70	H56	C34	C60	C62
0.78	1.97	-0.2	0.10	0.19	2.0	-1.0	2	-0.2

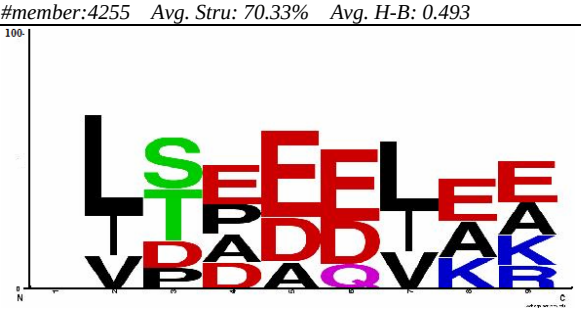
Level 1 Super-Rule 66
#member:7505 Avg. Stru: 65.89% Avg. H-B: -0.061



H47	H53	H65	H69	H70	H73	H74	H73	H69
2	-0.1	-1.0	-0.7	0.06	0.75	-0.6	-1.0	0.0

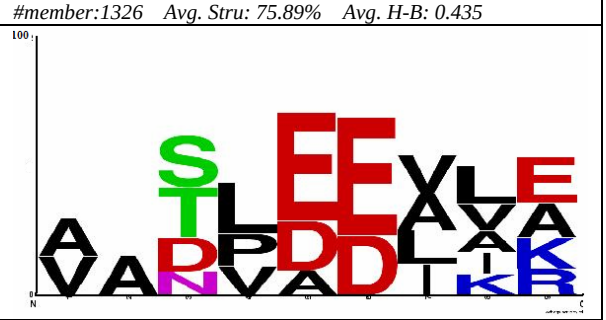
Table 6.3 Level 2 Super-Rule 2 (CH motifs)

Level 1 Super-Rule 19
#member:4255 Avg. Stru: 70.33% Avg. H-B: 0.493



C56	C66	C77	H61	H72	H76	H76	H75	H74
0	1.76	-0.2	-0.6	0.37	1.63	1.86	-0.3	-0.1

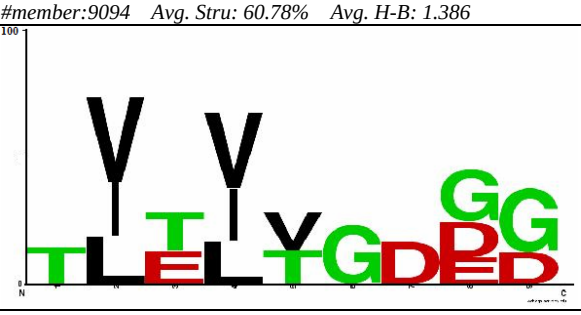
Level 1 Super-Rule 50
#member:1326 Avg. Stru: 75.89% Avg. H-B: 0.435



C61	C75	C79	H66	H78	H81	H83	H81	H79
0.0	2	0.31	-1.3	0.60	2.0	0.64	-0.2	-0.1

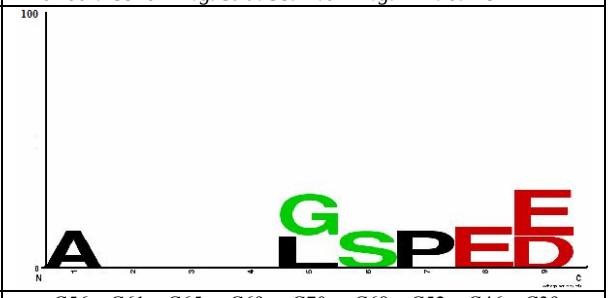
Table 6.4 Level 2 Super-Rule 3 (C, EC, CEC, ECE motifs)

Level 1 Super-Rule 20
#member:9094 Avg. Stru: 60.78% Avg. H-B: 1.386



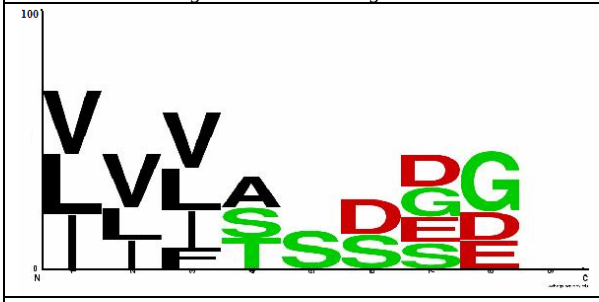
E52	E69	E75	E73	E60	E31	C59	C65	C63
2	2.05	-1.0	2.06	0.0	6	3	-0.6	-1.0

Level 1 Super-Rule 24
#member:13946 Avg. Stru: 58.44% Avg. H-B: 0.778



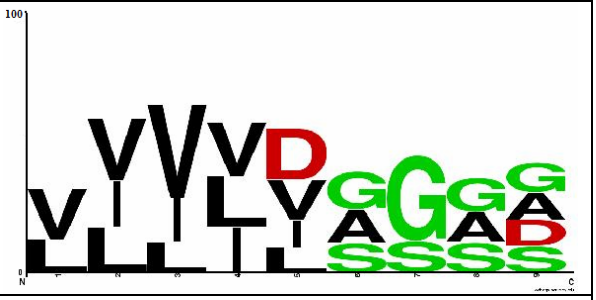
C56	C61	C65	C69	C70	C68	C52	C46	C39
2	0	0	0	-4.0	2	3	2	2.0

Level 1 Super-Rule 27
#member:6503 Avg. Stru: 62.44% Avg. H-B: 0.756



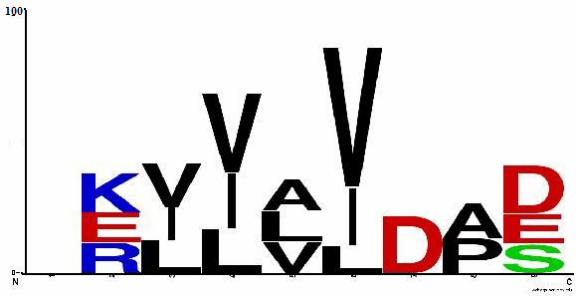
E69	E74	E71	E58	C38	C62	C68	C65	C57
1.97	1.96	1.17	0.68	2	0.0	-0.2	-0.8	0

Level 1 Super-Rule 33
#member:5666 Avg. Stru: 59.67% Avg. H-B: 0.68



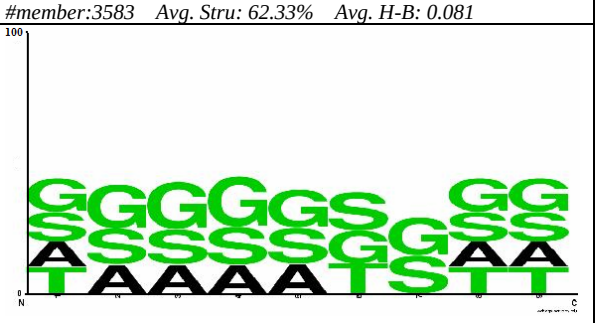
E51	E70	E75	E70	E49	C35	C64	C64	C59
1.0	1.99	2.12	1.98	-1.2	0.3	0.0	0.32	-0.33

Level 1 Super-Rule 34
#member:2359 Avg. Stru: 59.11% Avg. H-B: 1.47



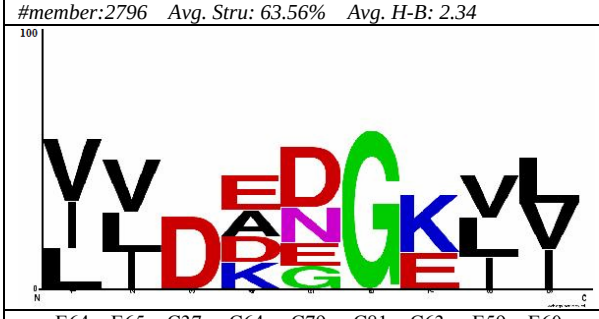
C60	C38	E57	E72	E79	E75	E61	E33	C57
0	1.08	1.99	2.10	0.0	2.29	6	-1.0	0.79

Level 1 Super-Rule 58
#member:3583 Avg. Stru: 62.33% Avg. H-B: 0.081



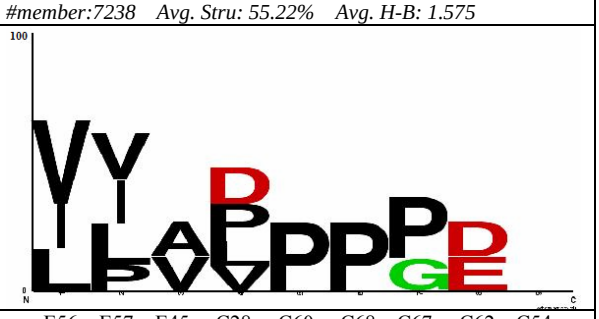
C54	C62	C67	C69	C69	C67	C63	C59	C51
-0.0	0.26	0.25	0.25	0.30	-0.2	0.0	-0.0	-0.0

Level 1 Super-Rule 62
#member:2796 Avg. Stru: 63.56% Avg. H-B: 2.34

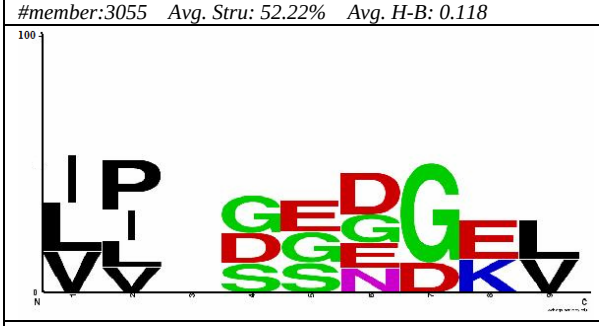


E64	E65	C37	C64	C79	C81	C63	E59	E60
2.03	1.96	6	-0.1	0.34	6	1.0	1.93	1.94

Level 1 Super-Rule 63
#member:7238 Avg. Stru: 55.22% Avg. H-B: 1.575



E56	E57	E45	C28	C60	C68	C67	C62	C54
2.02	0.24	0.0	-2.0	7	7	-2.0	2	0

Level 1 Super-Rule 65 *#member:3055 Avg. Stru: 52.22% Avg. H-B: 0.118*								
								
E42	E43	C33	C47	C68	C73	C71	C59	C34
2.0	-1.0	0	-0.4	-0.7	0.09	-1.0	1.0	1.0

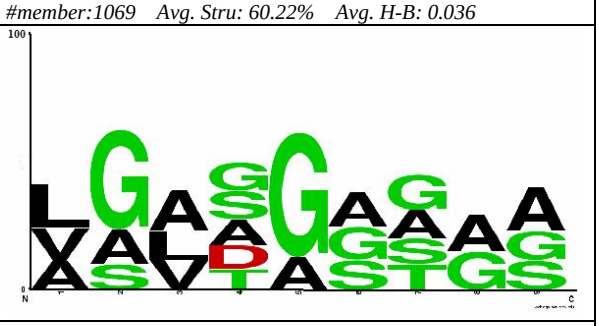
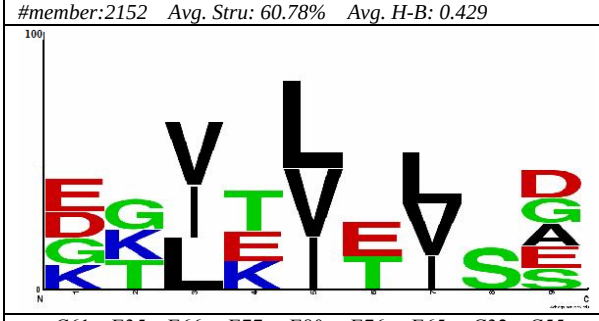
Level 1 Super-Rule 68 *#member:1069 Avg. Stru: 60.22% Avg. H-B: 0.036*								
								
C24	C56	C64	C73	C78	C73	C65	C56	C50
0.04	0.14	-0.1	-0.3	0.0	0.32	-0.0	0.0	0.4

Table 6.5 Level 2 Super-Rule 4 (CEC motifs)

Level 1 Super-Rule 25 *#member:2152 Avg. Stru: 60.78% Avg. H-B: 0.429*								
								
C61	E35	E66	E77	E80	E76	E65	C32	C55
-0.4	-1.7	2.0	-0.5	1.85	-1.0	1.89	2	-0.3

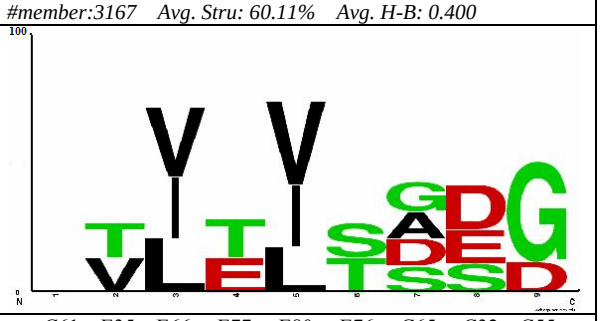
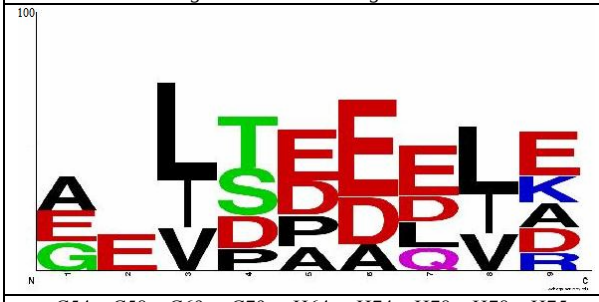
Level 1 Super-Rule 37 *#member:3167 Avg. Stru: 60.11% Avg. H-B: 0.400*								
								
C61	E35	E66	E77	E80	E76	C65	C32	C55
0	0.0	2.04	-1.0	2.10	1.0	-0.3	0.79	-1.0

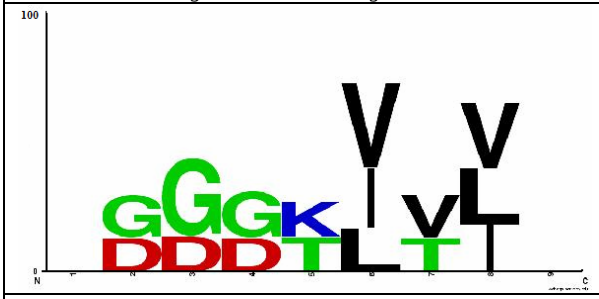
Table 6.6 Level 2 Super-Rule 5 (CH motifs)

Level 1 Super-Rule 28 *#member:2134 Avg. Stru: 69.89% Avg. H-B: 0.432*								
								
C54	C58	C69	C79	H64	H74	H78	H78	H75
-0.9	2	1.79	-0.3	-0.4	0.37	-0.4	1.85	-0.1

Level 1 Super-Rule 49 *#member:2267 Avg. Stru: 69.11% Avg. H-B: 1.589*								
								
C59	C65	C72	C79	H60	H71	H74	H72	H70
1.0	6	1.83	0.34	-1.8	2.0	2.0	1.08	1.83

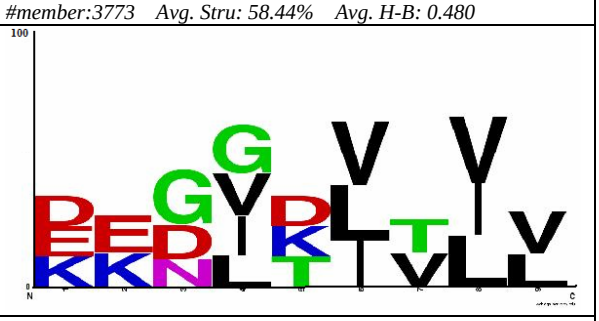
Table 6.7 Level 2 Super-Rule 6 (CE motifs)

Level 1 Super-Rule 30
#member:6226 Avg. Stru: 62.78% Avg. H-B: 0.008



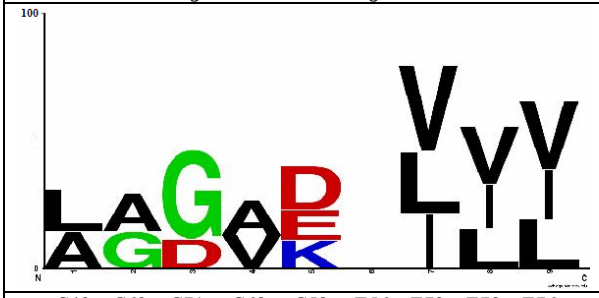
C53	C67	C71	C64	E39	E68	E73	E70	E60
0	-1.0	-1.0	-1.0	-1.0	2.13	0.0	1.94	0.0

Level 1 Super-Rule 36
#member:3773 Avg. Stru: 58.44% Avg. H-B: 0.480



C45	C64	C64	C36	E33	E72	E75	E74	E63
0.72	1.0	-0.2	-1.1	-1.0	1.94	0.0	2.01	0.0

Level 1 Super-Rule 52
#member:6022 Avg. Stru: 65.56% Avg. H-B: 0.511

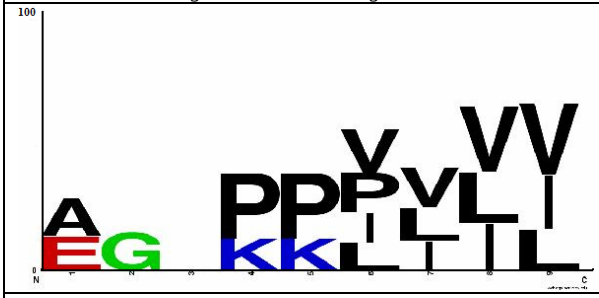


C49	C62	C71	C68	C58	E56	E72	E78	E76
-1.0	0.0	-1.0	0.0	0.62	0	1.96	2.01	2.01

Level 1 Super-Rule 53
#member:3480 Avg. Stru: 63.89% Avg. H-B: 0.958

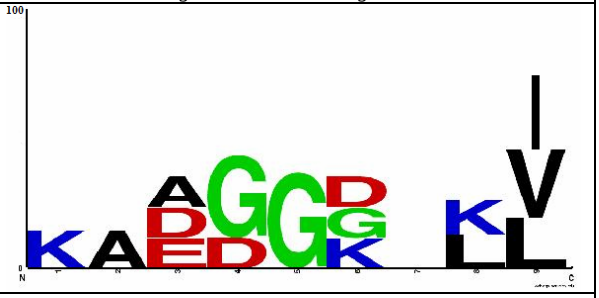
H78	H75	H69	C56	C81	C76	C38	E44	E58
-1.0	0.08	-0.3	-1.0	6	0.99	-0.2	1.94	2.07

Level 1 Super-Rule 59
#member:5945 Avg. Stru: 58.56% Avg. H-B: 0.625



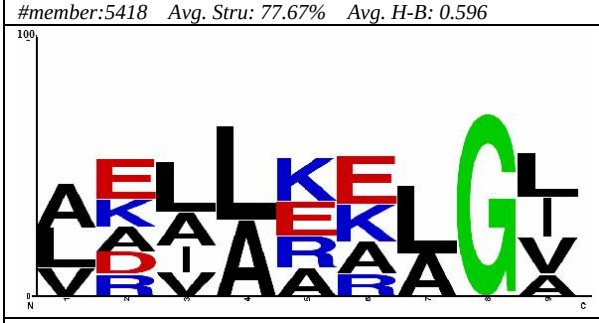
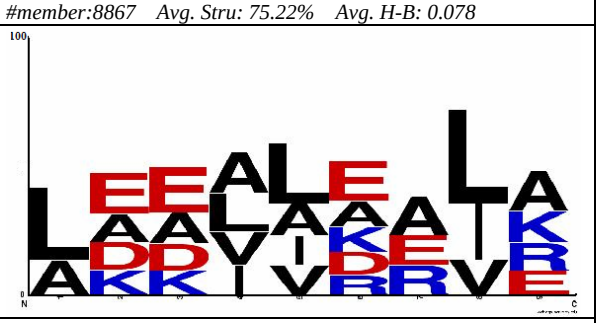
C54	C62	C70	C73	C63	C32	E47	E63	E63
-1.0	3	0	-1.0	-1.0	-0.4	1.96	1.98	2.10

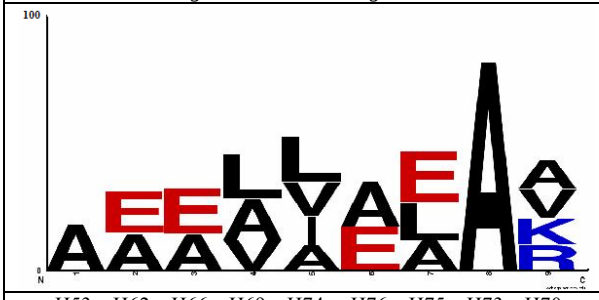
Level 1 Super-Rule 60
#member:3357 Avg. Stru: 52.89% Avg. H-B: 0.831



C29	C51	C60	C74	C71	C46	E34	E50	E61
2	2	-0.3	-1.0	6	-1.3	0	-2.0	2.13

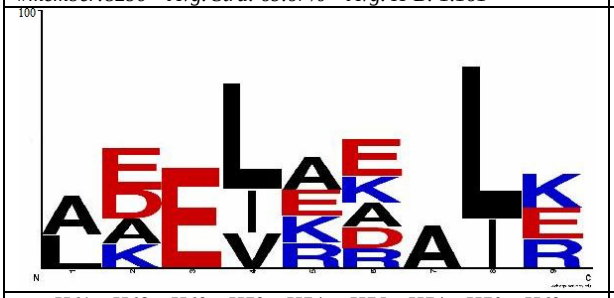
Table 6.8 Level 2 Super-Rule 7 (H motifs)

Level 1 Super-Rule 7 *#member:5418 Avg. Stru: 77.67% Avg. H-B: 0.596*	**Level 1 Super-Rule 31** *#member:8867 Avg. Stru: 75.22% Avg. H-B: 0.078*
	
H84 H86 H86 H84 H82 H76 C43 C81 C77	H84 H86 H86 H84 H82 H76 H43 H81 H77
-0.2 -0.1 0.49 -1.0 0.15 0.01 -1.0 6 1.0	-1.0 -0.1 -0.1 0.49 0.60 -0.1 -0.7 1.87 -0.2

Level 1 Super-Rule 39 *#member:6756 Avg. Stru: 68.67% Avg. H-B: 0.329*	**Level 1 Super-Rule 41** *#member:2622 Avg. Stru: 70.56% Avg. H-B: 0.727*
	
H53 H62 H66 H69 H74 H76 H75 H73 H70	H75 H78 H79 H78 H77 H74 H69 H28 C77
4 -1.0 -1.0 -0.1 0.80 -1.0 -1.8 4 -0.9	0.66 -0.1 -0.2 1.92 -1.0 0.14 -0.1 -0.8 6

Level 1 Super-Rule 42 *#member:7052 Avg. Stru: 77.67% Avg. H-B: 0.200*	**Level 1 Super-Rule 43** *#member:4108 Avg. Stru: 74.89% Avg. H-B: 0.057*
	
H71 H78 H80 H82 H82 H82 H80 H75 H69	H60 H66 H69 H78 H81 H83 H82 H80 H75
-1.7 0.05 -1.0 0.68 0.07 0.05 1.89 1.88 -0.1	-1.0 -0.1 -1.0 1.94 -0.1 -0.1 -1.0 1.93 -0.0

Level 1 Super-Rule 46
#member:8290 Avg. Stru: 69.67% Avg. H-B: 1.161

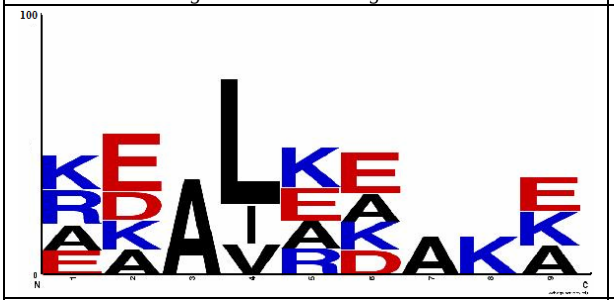


H61	H68	H69	H73	H74	H75	H74	H70	H63
-1.0	-0.1	5	1.75	-0.1	-0.1	2	2.0	1.05

Level 1 Super-Rule 47
#member:5939 Avg. Stru: 39.33% Avg. H-B: 0.268

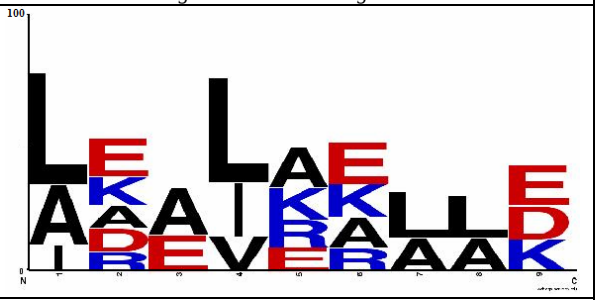
H61	H68	C69	C73	H74	H75	H74	E70	E63
-1.3	-0.1	-1.0	-1.9	2	0.78	0	1.99	1.95

Level 1 Super-Rule 48
#member:3822 Avg. Stru: 55.78% Avg. H-B: 1.064



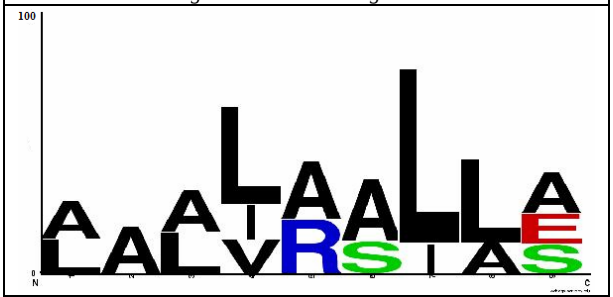
H61	H68	H69	H73	H74	H75	H74	C70	C63
0.17	0.03	4	1.69	0.12	-0.1	2	2	-0.3

Level 1 Super-Rule 54
#member:4330 Avg. Stru: 66.67% Avg. H-B: -0.092



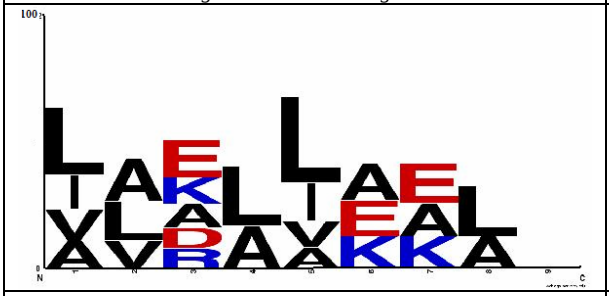
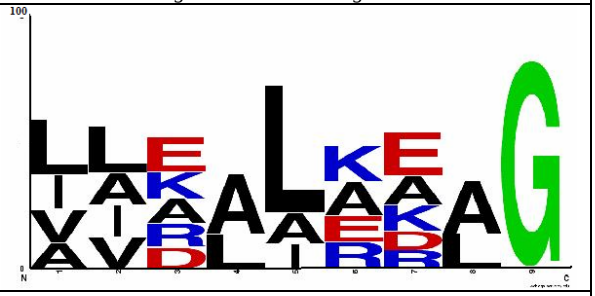
H79	H80	H78	H77	H74	H68	H61	H44	H39
-0.2	-0.1	-1.0	1.83	-0.1	-0.0	-1.0	-1.0	0.77

Level 1 Super-Rule 55
#member:6890 Avg. Stru: 63.33% Avg. H-B: 0.526



H58	H61	H62	H68	H69	H68	H66	H61	H57
-1.0	4	-1.0	1.74	-1.0	1.0	2.0	-1.0	0.0

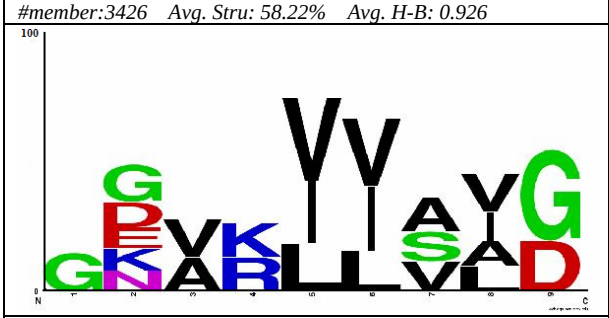
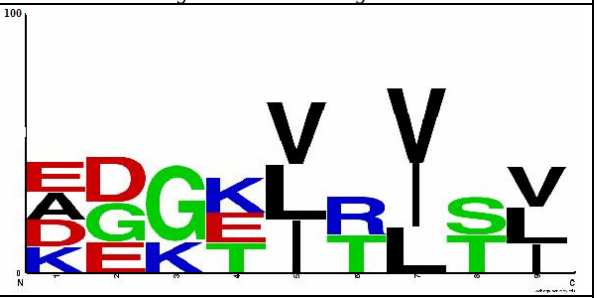
Table 6.9 Level 2 Super-Rule 8 (HC motifs)

Level 1 Super-Rule 32 *#member:17613 Avg. Stru: 61.22% Avg. H-B: -0.144*	**Level 1 Super-Rule 61** *#member:3414 Avg. Stru: 76.22% Avg. H-B: 0.602*
	

H76	H78	H77	H75	H70	H60	H50	H31	C34
0.74	-0.1	-0.1	-1.0	0.88	-0.4	-0.3	-1.0	0

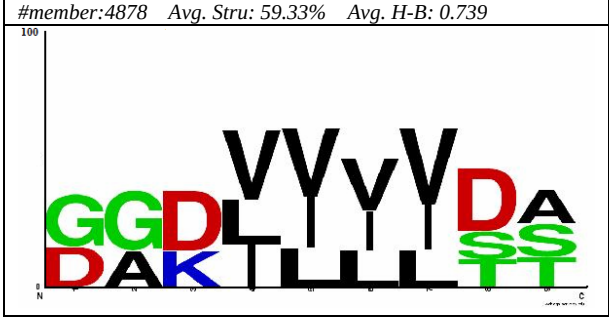
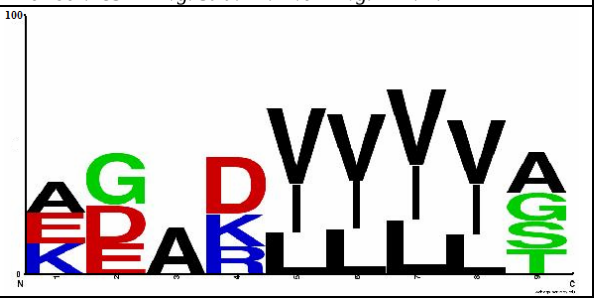
H78	H81	H82	H81	H79	H75	H69	C60	C81
0.86	0.67	-0.2	-1.0	0.20	0.01	-0.2	-1.0	6

Table 6.10 Level 2 Super-Rule 9 (CE motifs)

Level 1 Super-Rule 35 *#member:3426 Avg. Stru: 58.22% Avg. H-B: 0.926*	**Level 1 Super-Rule 38** *#member:3096 Avg. Stru: 61.44% Avg. H-B: 0.345*
	

C66	C66	C49	E35	E70	E73	E68	E45	C52
3	-0.4	0	2.0	2.10	2.15	-0.3	0.79	-1.0

C61	C69	C68	C29	E64	E73	E75	E67	E47
-0.3	-0.3	-2.0	-0.3	1.97	-1.0	2.11	1.0	1.93

Level 1 Super-Rule 40 *#member:4878 Avg. Stru: 59.33% Avg. H-B: 0.739*	**Level 1 Super-Rule 57** *#member:2831 Avg. Stru: 71.11% Avg. H-B: 1.177*
	

C67	C65	C34	E65	E75	E76	E69	E41	C52
-1.0	0.0	-1.0	1.98	2.10	2.02	2.09	-0.2	-0.68

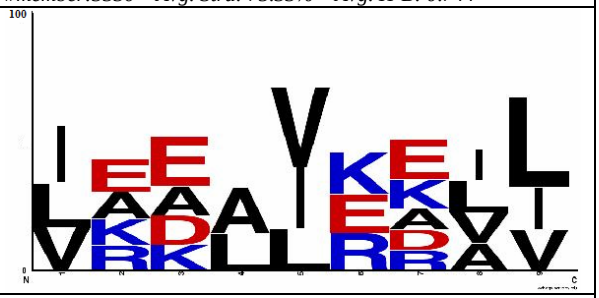
C65	C74	C73	C54	E68	E82	E85	E79	E60
-0.3	-0.6	4	-0.7	2.06	2.02	2.01	2.09	0.08

Table 6.11 Level 2 Super-Rule 10 (H motifs)

Level 1 Super-Rule 44
#member:4483 Avg. Stru: 77.00% Avg. H-B: 0.539

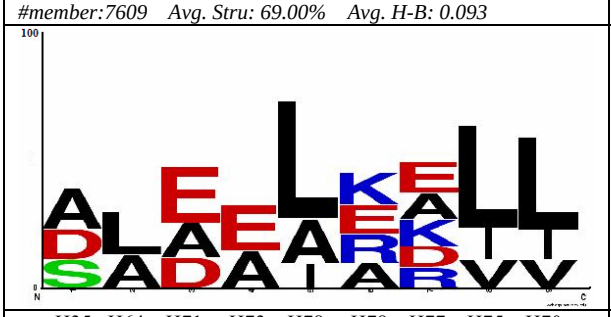
C76	H63	H73	H78	H82	H83	H82	H80	H76
0.08	-0.6	0.33	2.0	0.59	-0.1	-0.1	1.88	0.71

Level 1 Super-Rule 45
#member:3330 Avg. Stru: 75.33% Avg. H-B: 0.744



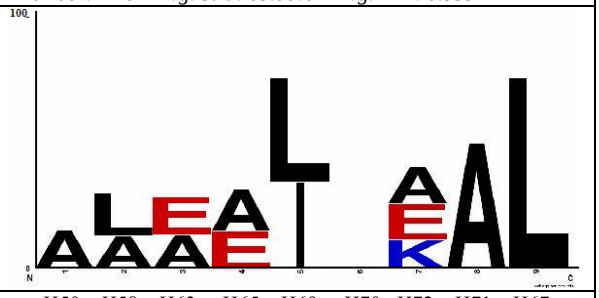
H69	H74	H79	H80	H79	H78	H76	H74	H69
2.07	0.03	-0.1	-1.0	2.14	1.01	-0.0	0.77	1.79

Level 1 Super-Rule 56
#member:7609 Avg. Stru: 69.00% Avg. H-B: 0.093



H35	H64	H71	H73	H78	H78	H77	H75	H70
-0.3	-1.0	-0.1	-1.0	0.02	0.07	-0.2	1.72	1.70

Level 1 Super-Rule 67
#member:4416 Avg. Stru: 65.00% Avg. H-B: 0.959



H50	H58	H63	H65	H69	H70	H72	H71	H67
2	-1.0	-1.0	-1.0	2.0	0	-0.4	4	4

CHAPTER 7

MINING POSITIONAL ASSOCIATION SUPER-RULE ON FIXED-SIZE PROTEIN SEQUENCE MOTIFS

As we discussed in the beginning of chapter 6, in order to identify motifs, a fixed window size technique is usually required. The fixed window size property usually cause two problems: the results may generate some similar motifs *(1) including mismatches* or *(2) shifted by some residues*. To solve the second problem, Jensen et al [14] proposed a clique enumeration approach to link up the clusters they found. Since clique enumeration is a NP-complete [50] problem, while dealing with large number of clusters, the computational cost will be high. We introduce a new algorithm called Positional Association Rule, with Apriori concept, to confront the second problem caused by fixed window size.

7.1 Association Rules

The notion of the association rule was proposed to capture the co-occurrence of items in transactions [51]. There are many improvements or advanced association rules algorithms [52-76]. An interesting extension to the association rule in the Bioinformatics field is to regard each DNA/protein sequence as a transaction and sequence motifs as items which appear in the transactions. A couple of papers apply association rules in this manner [77, 78]. One special property occurs if we treat DNA/protein sequence as a transaction: the transaction itself exists as an ordering relation. Therefore, if we try to map the protein sequence motifs (items) onto the

protein sequences (transactions), we may obtain the order of the occurring motifs. Icev *et al*[77] have successfully incorporated this idea into their Distance-Enhanced association rules algorithm. They use the coefficient of variation of distances (cvd) [79] concept to determine whether similar distances occur among pair of motifs. The cvd of a pair of motifs with respect to a collection of motifs is the ratio between the standard deviation and the mean of the distances between the motifs in those promoter regions that contain all the motifs [77]. However, to the best of our knowledge, there is no association rule algorithm that has the ability to tell what the exact distance among frequent itemsets is. For example, if motif A occurs, after 3 positions, motif B occurs. In this chapter, we propose a new *Positional Association Rule*s algorithm to search not only the frequent itemsets but also the distance between them.

7.2 Positional Association Rule Algorithm

The basic association rule gives the information of A => B. However, under the circumstances of the "order" involved with the appearance of items, the basic association rule is not powerful enough. For example, in our research, the information we need is not only limited in "If motif A appears in a protein sequence, then motif B also appears" but also considers "the information of the distance between motif A and B in the sequence." Therefore, the conventional support and confidence is not enough. In this chapter, we introduce two additional parameters called "*distance support and distance confidence*" to help identify frequent item with frequent distance, and it is applied after the strong association rules are obtained, rules that pass the check of minimum support and confidence.

To define the Positional Association Rule formally, I = {Ii, i=1,…, m} be a set of items. A transaction database *T* is a set of transactions, where each transaction *t* is a set of items that $t \subseteq I$. A positional association rule is an implication of the form $X \Rightarrow Y$ and $X \overset{d}{\Rightarrow} Y$, where *X* and *Y* both belong to *I*, and $X \cap Y$ equals empty. The support and confidence for an association rule $X \Rightarrow Y$ are defined as follows:

$$Support(X \Rightarrow Y) = \frac{|X \cup Y|}{|T|} \tag{7.1}$$

$$Confidence(X \Rightarrow Y) = \frac{|X \cup Y|}{|X|} \tag{7.2}$$

Where |*T*| is the total number of transactions, |*X*| is the number of transactions in *T* that contains at least one *X*, $|X \cup Y|$ is the number of the transactions in *T* that contain both *X* and *Y*. The newly proposed "*distance support*" and "*distance confidence*" is defined as:

$$Dis.Support(X \overset{d}{\Rightarrow} Y) = \frac{\left\| X \overset{d}{\cup} Y \right\|}{|T|}$$

$$Dis.Confidence(X \overset{d}{\Rightarrow} Y) = \frac{\left\| X \overset{d}{\cup} Y \right\|}{\|X\|} \tag{7.3}$$

Where ||*X*|| is the total number of times that X appears in *T, d* indicates the distance, $-\infty < d < \infty$. Where $X \overset{d}{\Rightarrow} Y$ denotes "if *X* appears, then after the distance of *d*, *Y* appears," $\|X \overset{d}{\cup} Y\|$ is the total number of times in *T* that when *X* occurs and after the distance of *d, Y* occurs. The problem of mining positional association rules lies in searching all positional association rules that have their value of support, confidence, distance support, and distance confidence higher than pre-defined minimum support, minimum confidence, minimum distance support and minimum distance confidence.

Algorithm: Positional Association Rule with the Apriori Concept
Input: Database, D, (Protein sequences as Transactions and Sequence Motifs as items), min_support, min_confidence, min_distance_support, and min_distance_confidence
Output: P, positional association rules in D.
Method:

```
(1)   L = find_frequent_itemsets(D, min_support)
(2)   S = find_strong_association_rules(L, min_confidence)
(3)   for (k=2; Sk ≠ Ø; k++)
(4)      for each strong association rule, r ∈ Sk
(5)         antecedent_motif = Apriori_Motif_Construct(r_ant)
(6)         consequence_motif = Apriori_Motif_Construct(r_con)
(7)         if antecident_motif == NULL or consequence_motif == NULL:
(8)            goto Step (4)
(9)         for each protein sequence, ps ∈ D
(10)           for (ant_position=1; |ps| ; ant_position++)
(11)              if antecedent_motif start appear on ps[ant_position]:
(12)                 r_ant_count++
(13)                 for (con_position=1; |ps| ; con_position++)
(14)                    if consequent_motif start appear on ps[con_position]:
(15)                       distance = ant_position – con_position
(16)                       r_distance ++
(17)        Pk = { r_distance | (r_distance > min_distance_support * num_ protein_sequence) and
                            (r_distance > min_distance_confidence * r_ant_count) }
```

Apriori_Motif_Construct(itemset)

```
(1)   if |itemset| == 1:
(2)      return itemset
(3)   else:
(4)      for each positional association rules in P|itemset|
(5)         if all items in the itemset appear in the positional association rule:
(6)            return the new motif constructed by the positional association rule
(7)      return NULL
```

Figure 7.1 The Pseudocode of Positional Association Rule with the Apriori concept

To compute the Positional Association Rule, we need to search frequent itemsets first. Thus, we need to compute support and confidence by the traditional association rule. After we identify

strong frequent itemsets, we work on looking for frequent distances between two frequent itemsets. We may consider each protein sequence as a transaction and each sequence motif as an item. Figure 7.1 shows the pseudocode for the Positional Association Rule with the Apriori concept.

Step 1 of the Positional Association Rule algorithm finds all of the frequent itemset, L, from the whole database. Step 2 discovers all strong association rules, S, which indicates the rules that pass both the minimum support and the minimum confidence check. Since the Apriori algorithm to find strong association rules is well developed [51, 53], we have skipped the details of how to find strong association rules in the Figure 2 to save space. In step 3 to 16, for each strong association rule, it tries to find if there exists a fixed distance between the antecedent and the consequent of the rules. The algorithm begins searching the strong association rules with two itemsets, S_2, to the maximum number of itemsets, S_k (Step 3). The idea is that we start exploring the specific distance in the strong association rules with a lower number of itemsets, then we search on the association rules with a higher number of itemsets; we may use the obtained information to facilitate the search process. For each association rule, r, in all S_k (step 4), the rules are in the form of antecedent_itemset => consequent_itemset. We use r_ant and r_con to represent those, respectively. The **Apriori_Motif_Construct** procedure is used for constructing new motifs if necessary and possible. It takes either r_ant or r_con as a parameter. If the input contains only one item (step 1), it indicates it is not necessary to generate a new motif information since it already existes; therefore, it returns input itself (step 2). If the input contains more than one item, it is necessary to construct a new motif (step 3) if possible. The program needs to search all positional association rules so that the sum of antecedent and consequent items is equal to the number of input items (step 4). If there is a match (step 5), the procedure

returns the new motif information based on the found positional association rule (step 6). Otherwise, the procedure returns NULL in the end of the search.

Step 5 to 8 of the main program is where Apriori concept is implemented. Step 5 and 6 call the **Apriori_Motif_Construct** procedure to obtain the motif information for r_ant and r_con. If either **Apriori_Motif_Construct(**r_ant**)** or **Apriori_Motif_Construct(**r_con**)** returns Null (step 7), this indicates that the positional association rule of either the antecedent or consequent of the rule can not be found; therefore, we may not treat either one of those as "one" single larger motif. Thus, we cannot find any specific distance for the rule we are working on, so we may test the next strong association rule (step 7). If both antecedent_motif and consequent_motif can be contracted, we search every position of all protein sequences (step 9 and 10). Once we find that the antecedent_motif appears starting on the protein sequence position that we are looking at, we increase the count of antecedent appearing by one (step 11 and 12). And then we search for the appearance of consequent motif on every position of the same protein sequence that we found antecedent motif (step 13). If the consequent occurs, we calculate the distance between antecedent and consequent of the rule and increase one count of the distance (step 14 – 16). After the scan of all protein sequences, if some distance counts are greater than the thresholds, which are the minimum distance support multiplied by the number input protein sequences and the minimum distance confidence multiplied by the number of time the antecedent occurs, we may include the distance to the strong association rule and form the Positional Association Rule (step 17). In order to give a clear explanation of the pseudocode, we include the following example.

7.3 An Example of Positional Association Rule Algorithm

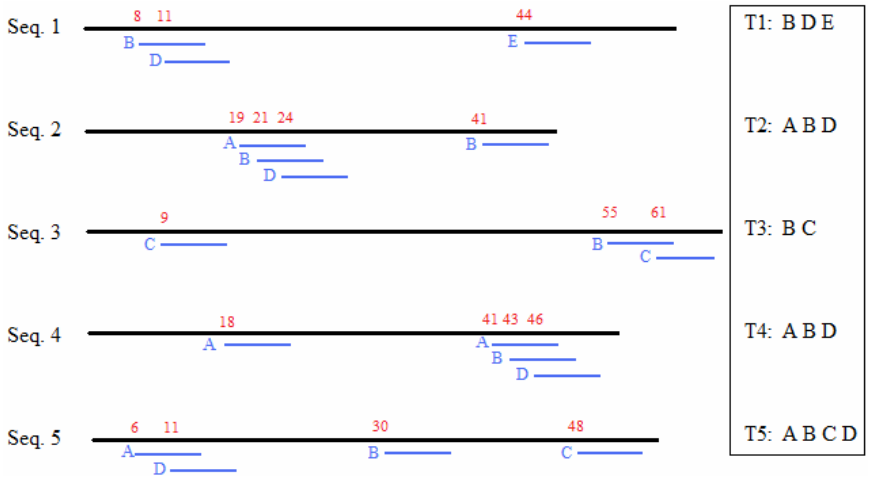


Figure 7.2 Example for Positional Association Rule search (minimum support = 60%, minimum confidence = 80%, minimum distance support = 40%, and minimum distance confidence = 60%)

Figure 7.2 demonstrates an example of five protein sequences with variable length and five different motifs (A, B, C, D, and E) with the fixed window size of 9. The numbers above the sequence indicate the starting location where the motifs have been identified.

We produce positional association rules from strong association rules. After the computation of the traditional association rule algorithm, we obtained strong association rules: {A=>B}, {A=>D}, {B=>D}, {D=>B}, {A=>BD}, {AD=>B} and {AB=>D}. We start on strong association rules with two itemsets, take {A=>D} as the first example. First of all, we need to find out the value of ||A||, the instances of A occuring in all transactions. By scanning the whole database, motif A occurs once in Seq. 1, Seq. 5 and twice in Seq. 4, we have ||A||=4. The second step is to find all different distances between motif A and D, Seq. 2, 4, and 5 yields {A=>D} with 5 distances, and Seq. 4 gives {A=>D} with 28 distances as well. The final step is to

calculate the distance support and confidence for all possible distances. Thus, we have $A \overset{5}{\Rightarrow} D$ for 3 times and $A \overset{28}{\Rightarrow} D$ once, and since |T| = 5 and ||A|| = 4, the distance support for $A \overset{5}{\Rightarrow} D$ = 60% and the distance confidence for $A \overset{5}{\Rightarrow} D$ = 75%. Therefore, based on the threshold we set, we have the information that "when motif A occurs, after 5 amino acid distances, motif B also occurs." Take {A=>B} for another example: we have $A \overset{2}{\Rightarrow} B$ twice, $A \overset{22}{\Rightarrow} B$ once, $A \overset{25}{\Rightarrow} B$ once, and $A \overset{24}{\Rightarrow} B$ once. Since none of them pass the minimum distance assurance check, we do not generate any positional association rule from strong association rule {A=>B}. In Figure 1, we can find two positional association rules in two itemsets that match the predefined criteria: $A \overset{5}{\Rightarrow} D$ and $D \overset{-3}{\Rightarrow} B$, where the second one can be explained as "when motif D occurs, before 3 amino acid distances, motif B also occurs."

After we discovered 2-itemsets positional association rules, we may proceed to 3-itemsets based on Apriori concept. Association rules information is in the form of A=>B, and we called A as *antecedent* and B as *consequent*. While an association rule contains more than three itemsets, either antecedent or consequent (or both) must contain more than one itemset. In Positional Association Rules algorithm, we need to make sure that the antecedent or the consequent contains more than one itemset, which needs to have at least one positional association rule among all of the itemsets it contains; this is where Apriori concept is applied. While we dealing with k-itemsets, we can always use k-1, k-2…, 2-itemsets positional association rules information to check if whether there is at least one positional among the antecedents or the consequents. For example, we have 3-itemsets strong association rules {AB=>D}, and since the antecedent contains more than one item, we need to check if there is a positional association rule among motif A and B. Unfortunately, there is not; therefore, we can not treat motif A and B as "one" larger motif and search the possible distances between AB and

D. On the other hand, if we take a look at the rule {AD=>B}, there does exist one positional association rule $A \overset{5}{\Rightarrow} D$; thus, we can treat motif A and D as a combined motif with length of 14 and search the possible distances connecting to motif B. In this example, the combined motif A and D with 14 lengths appears in Seq. 2, 4 and 5, so ||AD||=3. The possible distances between AD and B are 2, 22, 24 found in Seq. 2 and 4, 4, 5, respectively. As a result, we may have $AD \overset{2}{\Rightarrow} B$ twice, $AD \overset{22}{\Rightarrow} B$ once, $AD \overset{24}{\Rightarrow} B$ once, based on the minimum distance assurance we defined, $AD \overset{2}{\Rightarrow} B$ is a 3-itemsets positional association rule. The last example is {A=>BD}, and since the consequent contains motifs B and D and we do have $D \overset{-3}{\Rightarrow} B$ rule, we can regard BD as a size 12 motif. Due to the fact that ||A||=4, and we have $A \overset{2}{\Rightarrow} BD$ twice, $A \overset{25}{\Rightarrow} BD$ once, and since none of them pass the distance confidence threshold, no positional association rule is generated from the last example.

Although the example in Figure 3 does not contain any 4-itemsets (or higher) strong association rules, the method to produce positional association rules can be easily derived. Under 4-itemsets situation, there are only two types of scenarios: 1. either the antecedent or the consequent contains three items and the remaining one item is left to the consequent or antecedent. 2. Both the antecedent and the consequent carry two items. No matter what, while we are dealing with 4-itemsets, we already have the positional association rules information of 2-itemsets and 3-itemsets. We can always use those to test whether we should treat the items in the antecedent or the consequent as a "larger" item. We believe it follows the concept of Apriori algorithm and because of this computation time can be saved dramatically.

7.4 Mapping the Motifs (Items) onto the Protein Sequences (Transactions)

In order to identify the motif appearance in protein sequences, we use the sliding window technique with window size nine on each protein sequence to match with the motifs. If the dissimilarity is lower than some thresholds, we know the motif appears on the position. The threshold we setup includes two criteria: 1. the value of dissimilarity should be no more than 540 after the computation of the equation (4); 2. The sliding window segment on the protein sequence should have more than 6 positions that match with the motif's representative secondary structure.

7.5 Mining Positional Association Super-Rules on Fixed-Size Protein Sequence Motifs

In order to obtain the DNA/protein sequence motifs information, fixing the length of sequence segments is usually necessary. However, two major problems may occur: several motifs may look alike with some mismatches, or similar motifs may have shifted some residues. In section IV A, we deal with the first problem by constructing a Super-Rule-Tree structure, and we confront the second one in this subsection. Since the average size of Helixes is 10 [8], we believe the situation of the second problem will likely happen in our dataset. This problem is the motivation for us to create and apply the Positional Association Rules algorithm.

We fade all 343 motifs into Positional Association Rules algorithm as items and match those onto 2710 sequences. However, the rules we obtained are mostly $A \overset{0}{\Rightarrow} B$. This result can be easily understood by look at Figure 6.4, the SRT structure. Many rules are similar to one another, while matching the motifs onto the sequences, motifs that are alike will all be marked on the same position. So when we try to find the positional association rules, the rules like "when motif

A appears, motif B also appears on the same position" occur in the majority. Another difficulty with association rules algorithm is that it is notorious for generating redundant rules. As a result, we came up with an idea; instead of mining positional association rules on all motifs, we fade in level-1 super-rules instead. It is like a pre-process step before running the algorithm in order to reduce the number of redundant rules and this may also solve the 0-distance problem.

We fade 69 level 1 super-rules as items into our positional association rules algorithm; consequently, we name this approach as Positional Association "Super-Rules." Different parameters may result in generating different numbers of rules. We tried several different combinations of minimum support, minimum confidence, minimum distance support, and minimum distance confidence. Since with four parameters in one experiment it is really hard to obtain a thorough analysis on optimal parameter setup, we change the value of minimum support, confidence, distance confidence and fix minimum distance support to 20%. In order to find the suitable combination of parameters, we proposed a new evaluation method called "*HSSP-BLOSUM62 GAIN*." The idea comes from the fact that when we use Positional Association Rule algorithm to link two or more motifs, we cannot gain or lose any secondary structure similarity on each position of the new motif. This is because the computation of secondary structure similarity on each position considers all participated members; therefore, the value simply equals the average value. On the other hand, while the motifs are linked together, the noticeable amino acids on each position are slightly changed. Under the circumstance of not sacrificing secondary structure similarity, we use the Positional Association Rules algorithm to expand fixed window size motifs and to try to maximize the increase of HSSP-BLOSUM62 value. Thus, the "*HSSP-BLOSUM62 GAIN*" is calculated by the increase of the new motif's overall HSSP-BLOSUM62 value minus the average of all participated motifs' overall HSSP-

BLOSUM62 value. Table 7.1 shows the relation between different parameters and the HSSP_BLOSUM62 GAIN on the generated positional association rules (both 2-item and 3-item positional association rules).

The first, second, and third column of Table 7.1 indicates the setup parameters of the minimum support, minimum confidence and minimum distance confidence, respectively. The fourth column shows the number of 2-item positional association rules that are generated based on the parameters. The fifth column gives the total HSSP-BLOSUM62 GAIN of all generated rules. The sixth column presents the average HSSP BLOSUM62 GAIN for each rule. The seventh through ninth columns show similar trends to columns four through six for 3-item positional association rules. Analyzing the data given in Table 7.1, we recognize that if we setup a higher threshold, the number of generated rules decrease and the quality of the rules increase. More importantly, under the same minimum support and confidence, if we give a more strict minimum distance confidence, the situation follows the same trend.

Since we do not wish for either situation, (1) the whole process generates only one or two top most quality rules or (2) the whole process generates hundreds of meaningless rules. The value of "Total HSSP-BLOSUM62 2.1 GAIN" is designed to find a trade-off between the numbers of generated positional association rules and the quality improvement. The higher Total HSSP-BLOSUM62 2.1 GAIN, the more meaningful positional association rules that are generated. Figures 7.3 to 7.8 are the interpretation of Table 7.1 in terms of total HSSP-BLOSUM62 2.1 GAIN and different parameter values. By analyzing the figures, we made note that the highest Total HSSP-BLOSUM 2.1 GAIN always happens when minimum distance assurance equals 70%.

Table 7.1 The relation between the parameters setup and HSSP-BLOSUM62 GAIN

Support	Confidence	Distance Confidence	number of 2-itemset Rules	Total HSSP-BLOSUM62 GAIN	Ave. HSSP-BLOSUM62 GAIN	number of 3-itemset Rules	Total HSSP-BLOSUM62 GAIN	Ave. HSSP-BLOSUM62 GAIN
7.5%	60%	40%	158	-16.32	-0.103	1258	-374.01	-0.297
		50%	81	-4.05	-0.05	383	-102.86	-0.269
		60%	38	4.86	0.128	79	-10.17	-0.129
		70%	19	5.96	0.314	33	-1.24	-0.038
		80%	8	4.47	0.559	6	0.62	0.103
		90%	2	3.39	1.695	0	0	0
	70%	40%	114	-9.32	-0.08	713	-213.56	-0.300
		50%	66	-0.87	-0.013	256	-71.47	-0.279
		60%	37	5.26	0.142	73	-7.44	-0.101
		70%	19	5.96	0.313	33	-1.24	-0.038
		80%	8	4.47	0.559	6	0.61	0.102
		90%	2	3.39	1.695	0	0	0
	80%	40%	47	1.01	0.02	162	-40.93	-0.253
		50%	38	3.7	0.097	98	-22.85	-0.233
		60%	28	5.84	0.209	52	-8.82	-0.170
		70%	18	6.1	0.339	28	-2.94	-0.105
		80%	8	4.47	0.559	6	0.62	0.103
		90%	2	3.39	1.695	0	0	0
10%	60%	40%	122	-21.56	-0.177	85	-285.22	-0.334
		50%	61	-6.75	-0.110	237	-75.45	-0.318
		60%	25	3.79	0.152	32	-2.82	-0.088
		70%	12	4.76	0.397	10	2.56	0.256
		80%	7	4.49	0.641	3	1.77	0.59
		90%	2	3.39	1.695	0	0	0
	70%	40%	91	-12.94	-0.142	505	-160.87	-0.319
		50%	49	-3.35	-0.068	149	-44.01	-0.295
		60%	25	3.79	0.152	30	-1.24	-0.041
		70%	12	4.76	0.397	10	2.56	0.256
		80%	7	4.49	0.641	3	1.77	0.59
		90%	2	3.39	1.695	0	0	0
	80%	40%	37	-0.63	-0.017	99	-23.76	-0.24
		50%	30	2.08	0.069	52	-11.09	-0.213
		60%	20	4.21	0.211	17	-0.47	-0.028
		70%	12	4.76	0.397	7	1.86	0.266
		80%	7	4.49	0.641	3	1.77	0.59
		90%	2	3.39	1.695	0	0	0
12.5%	60%	40%	99	-19.96	-0.202	491	-174.8	-0.356
		50%	48	-5.08	-0.106	157	-51.26	-0.327
		60%	20	3.4	0.17	25	-2.07	-0.083
		70%	9	4.12	0.458	8	1.51	0.189
		80%	6	4.49	0.748	3	1.77	0.59
		90%	2	3.39	1.695	0	0	0
	70%	40%	73	-10.82	-0.148	257	-78.71	-0.306
		50%	38	-1.374	-0.036	91	-20.35	-0.224
		60%	20	3.4	0.17	23	-0.49	-0.021
		70%	9	4.12	0.458	8	1.51	0.189
		80%	6	4.49	0.748	3	1.77	0.59
		90%	2	3.39	1.695	0	0	0
	80%	40%	29	-0.37	-0.013	48	-9.39	-0.196
		50%	23	2.52	0.110	35	-6.94	-0.198
		60%	16	3.98	0.249	14	0.13	0.009
		70%	9	4.12	0.458	7	1.86	0.266
		80%	6	4.49	0.748	3	1.77	0.59
		90%	2	3.39	1.695	0	0	0

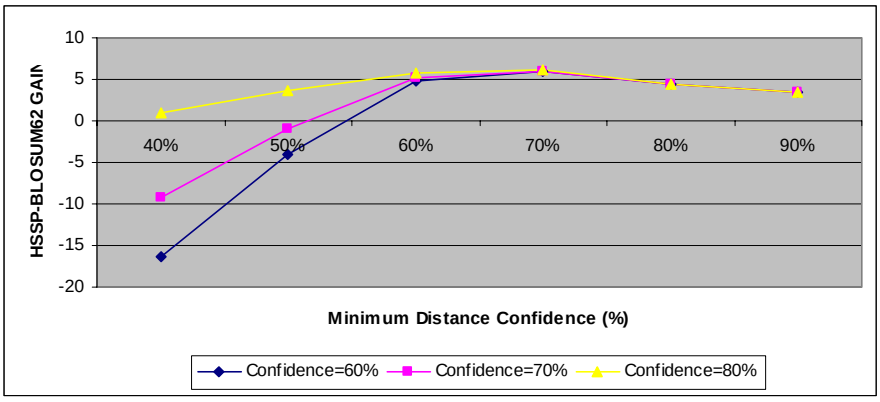


Figure 7.3 The relation between Total HSSP-BLOSUM62 GAIN and different parameter setup (minimum confidence and minimum distance confidence) when minimum support equals 7.5% on 2-itemset positional association rules results.

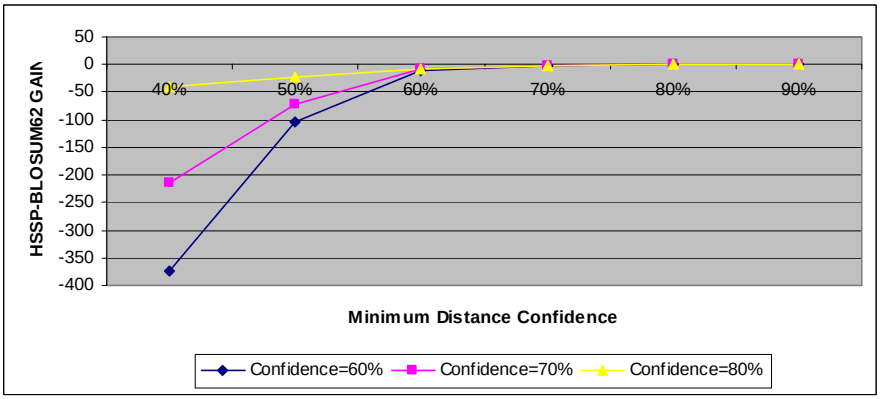


Figure 7.4 The relation between Total HSSP-BLOSUM62 GAIN and different parameter setup (minimum confidence and minimum distance confidence) when minimum support equals 7.5% on 3-itemset positional association rules results.

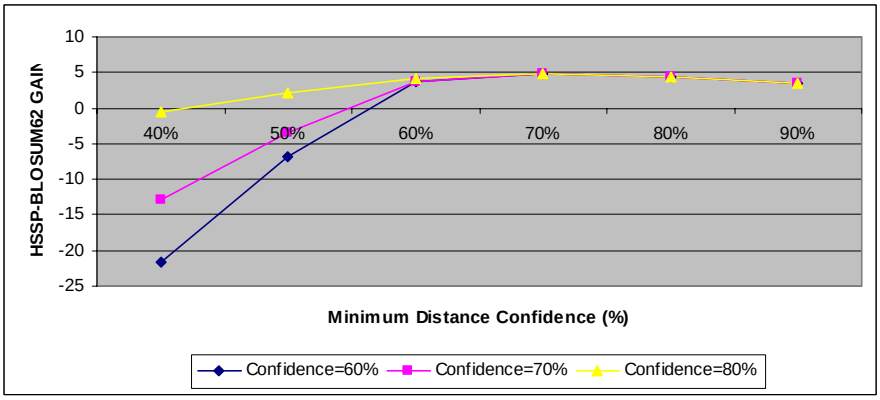


Figure 7.5 The relation between Total HSSP-BLOSUM62 GAIN and different parameter setup (minimum confidence and minimum distance confidence) when minimum support equals 10% on 2-itemset positional association rules results.

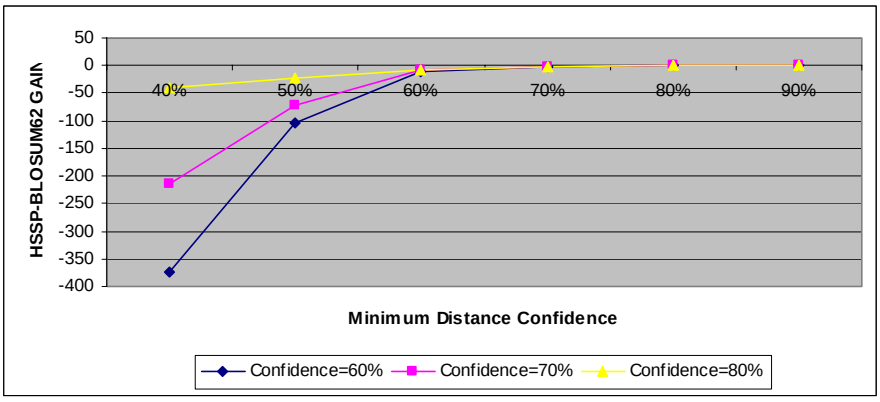


Figure 7.6 The relation between Total HSSP-BLOSUM62 GAIN and different parameter setup (minimum confidence and minimum distance confidence) when minimum support equals 10% on 3-itemset positional association rules results.

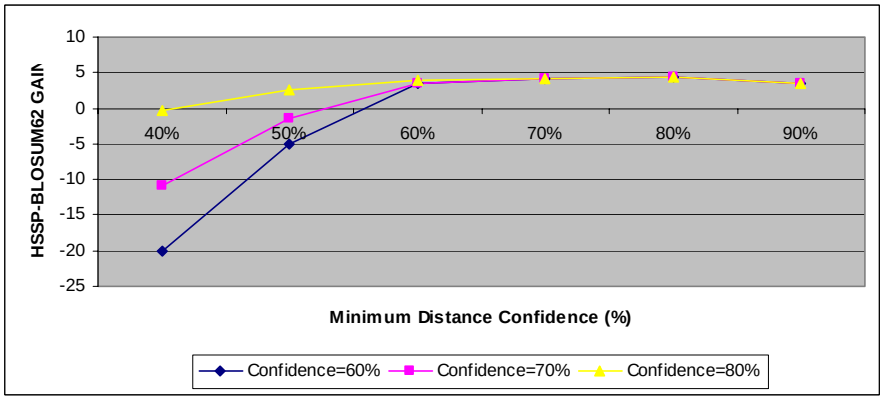


Figure 7.7 The relation between Total HSSP-BLOSUM62 GAIN and different parameter setup (minimum confidence and minimum distance confidence) when minimum support equals 12.5% on 2-itemset positional association rules results.

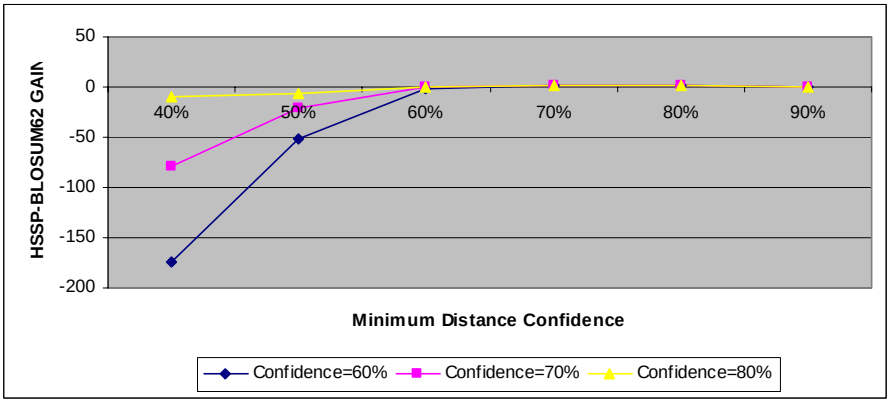


Figure 7.8 The relation between Total HSSP-BLOSUM62 GAIN and different parameter setup (minimum confidence and minimum distance confidence) when minimum support equals 12.5% on 3-itemset positional association rules results.

In order to understand the effect on the minimum support setup, we fix the minimum confidence and distance confidence and change minimum support from 5% to 15%. Table 7.2 gives the experimental results of this change and Figure 7.9, 7.10 is the interpretation of Table 7.2. We can see that for 2-itemset rules, higher support generates fewer numbers of rules with higher Average HSSP-BLOSUM62 GAIN. Nevertheless, since 3-itemset rules link three different motifs, which is a more complicated situation, it shows a peak on the Average HSSP-BLOSUM62 GAIN while minimum support equals 11%.

Table 7.2 The relation between the different minimum support setup and HSSP-BLOSUM62 GAIN

Support	Confidence	Distance Confidence	number of 2-itemset Rules	Total HSSP-BLOSUM62 GAIN	Ave. HSSP-BLOSUM62 GAIN	number of 3-itemset Rules	Total HSSP-BLOSUM62 GAIN	Ave. HSSP-BLOSUM62 GAIN
5%	60%	70%	24	6.79	0.283	37	0.43	0.012
6%			22	6.75	0.307	33	-1.24	-0.038
7%			19	5.96	0.313	33	-1.24	-0.038
8%			19	5.96	0.313	28	-1.41	-0.052
9%			17	5.91	0.348	17	1.79	0.105
10%			12	4.76	0.397	10	2.56	0.256
11%			10	4.03	0.403	9	2.87	0.319
12%			10	4.03	0.403	8	1.51	0.189
13%			9	4.12	0.458	8	1.51	0.189
14%			9	4.12	0.458	7	1.60	0.229
15%			7	3.99	0.570	0	0	0

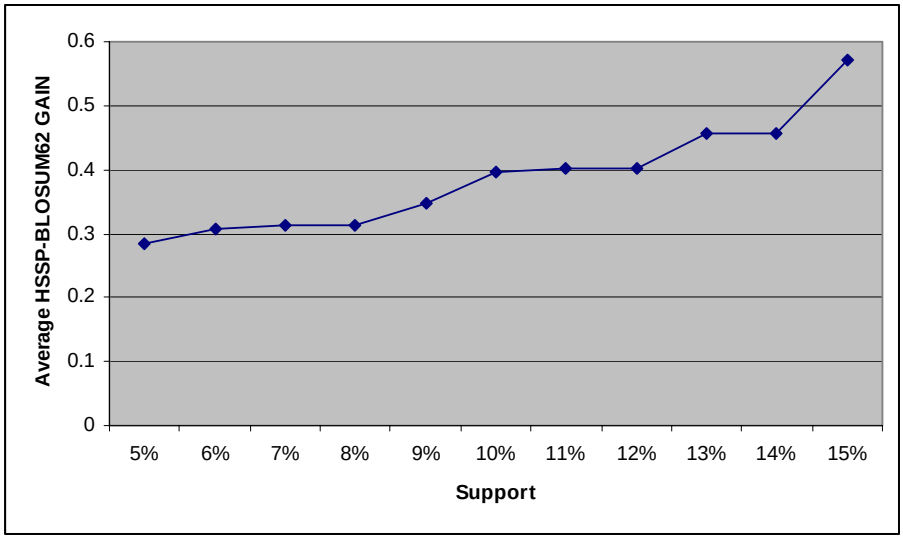


Figure 7.9 The relation between Average HSSP-BLOSUM62 GAIN and different minimum support setup on 2-itemset position association rules

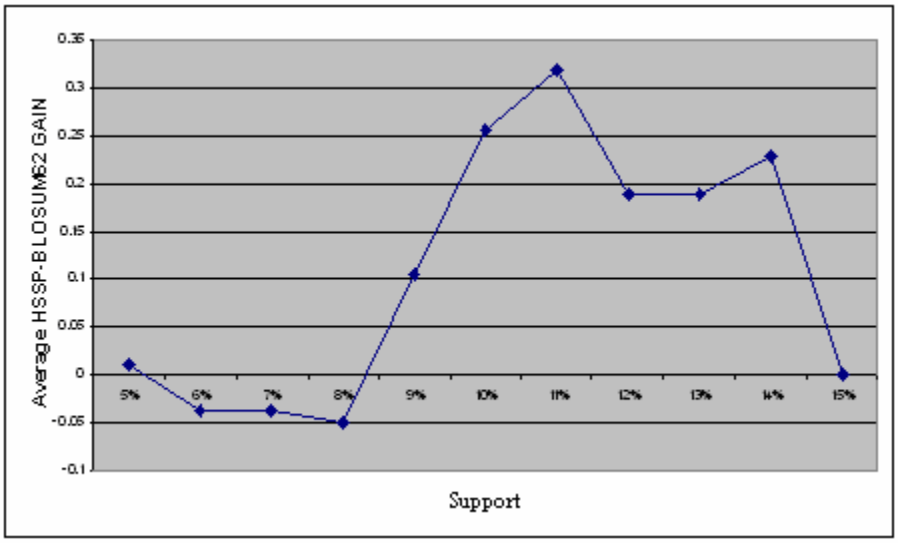


Figure 7.10 The relation between Average HSSP-BLOSUM62 GAIN and different minimum support setup on 3-itemset position association rules

7.6 Positional Association Super-Rules Example

Based on our experimental results, we show some positional association rules in this section. The following format is used in the 2-itemset example:

- The first row shows the positional association rule
- The second row gives the Support, Confidence, and Distance Assurance of the rule.
- The third row illustrates the antecedent motif of the positional association rule.
- The fourth row illustrates the consequent motif of the positional association rule.
- The last row gives a picture of the positional association rule's final product.

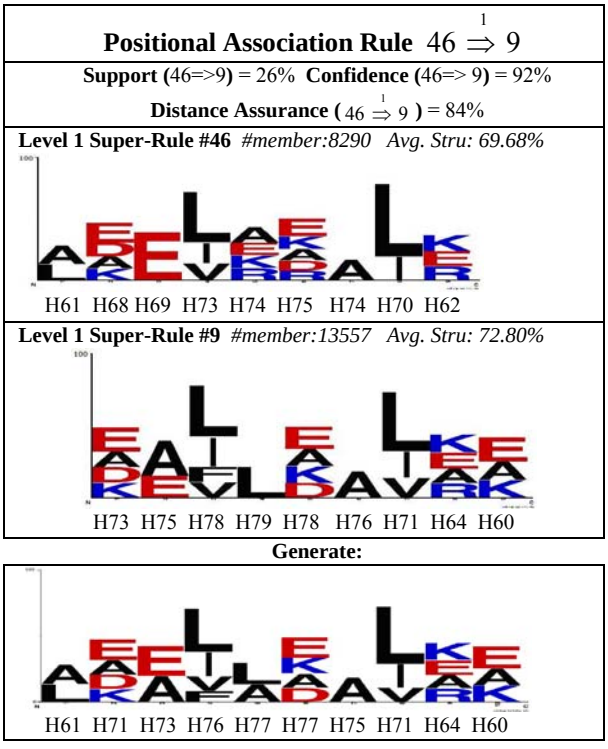


Figure 7.11 Positional Association Rule $46 \overset{1}{\Rightarrow} 9$

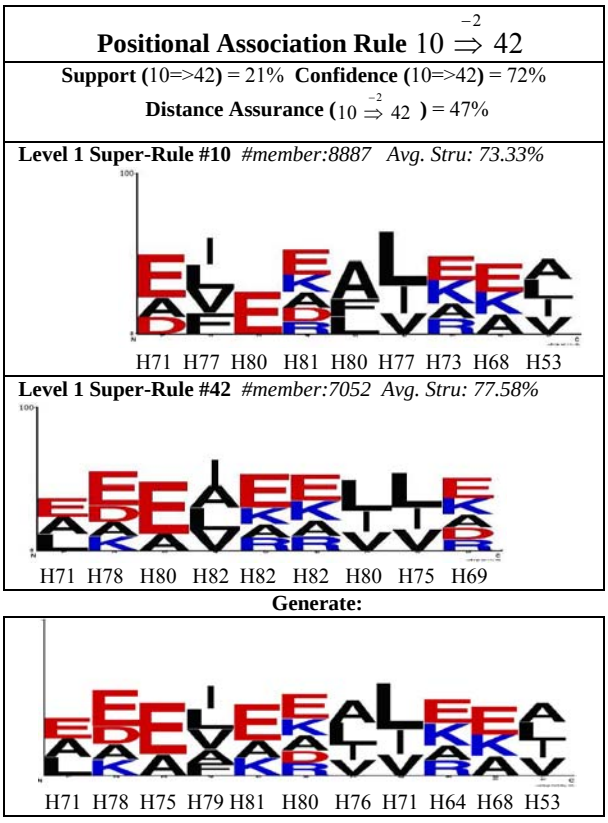


Figure 7.12 Positional Association Rule $10 \overset{-2}{\Rightarrow} 42$

The following format is used in the 3-itemset example:

- The first row shows the positional association rule.
- The second row gives the Support, Confidence, and Distance Assurance of the rule.
- The third and fourth row illustrates the antecedent motif of the positional association rule.
- The fifth row illustrates the consequent motif of the positional association rule.
- The last row gives a picture of the positional association rule's final product.

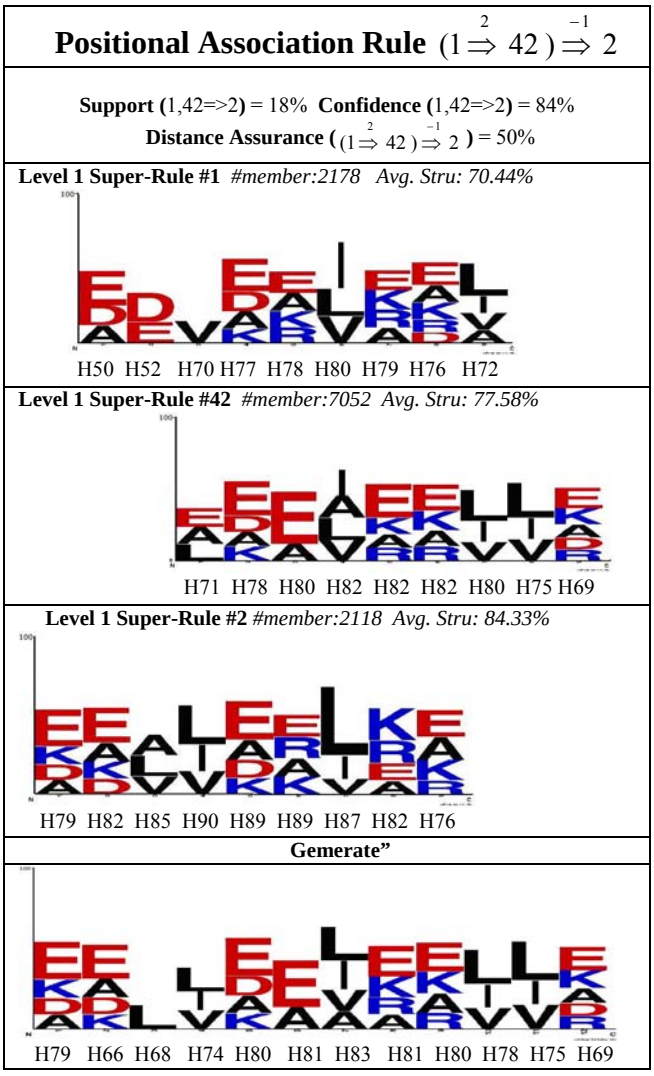


Figure 7.13 Positional Association Rule $(1 \overset{2}{\Rightarrow} 42) \overset{-1}{\Rightarrow} 2$

7.7 Conclusion

In both chapters 6 and 7, we propose several novel ideas: (1) we modified Hybrid Hierarchical K-means clustering algorithm into a parameter-free approach and applied it to construct the Super-Rule-Tree (SRT). (2) SRT is a higher level of super rule concept; it can serve as a visualized graph analysis of the similarity of motifs. (3) More importantly, we propose a novel

association rules scheme named Positional Association Rules Algorithm. Distance support and distance confidence are created for searching frequent distance among frequent itemsets. (4) We combine Positional Association Rule algorithm with super-rule concept by feeding level one super-rules as the input to confront the problems caused by fixed window sizes. (5) We also modify the HSSP-BLOSUM62 measure for biochemical evaluation in a more reasonable way and offer the HSSP-BLOSUM62 GAIN evaluation on parameter setup. After a series of experiments, we give a detailed analysis of data and conclude the optimal parameters setup. Although the works described in this chapter are merely based on protein primary structure computation, the results are also meaningful to protein secondary structure as well as the biochemical point of view.

CHAPTER 8

DISCOVERING PROTEIN SEQUENCE MOTIF THROUGH HIGH PERFORMANCE COMPUTING

8.1 Motivation

Although both the FIK and FGK model reduce the execution time to only 20% of the original required, it still takes more than 17 days to finish the whole program in our experiment, in which all codes are written in Python. In this case, it is very difficult to obtain different motif information with various window sizes. Since Python is a script language, the execution time is usually much slower than traditional languages such as C/C++ or JAVA. Therefore, we rewrite the whole program in C to see a difference. Furthermore, since C support MPI for multiprocessor execution, we carry out the parallelization step for our protein sequence motif research.

8.2 Parallel K-means Clustering Algorithm

The parallel K-means design is based on message passing interface (MPI) on a distributed memory system. MPI is a robust, efficient, portable, and friendly using on C/C++. In this chapter, all codes are based on C. The basic idea for parallel K-means clustering algorithm is to evenly distribute the data in master processor to slave processor in order to find out to which cluster each data point belongs. The initial centroid locations are generated by random in master

processor and then broadcast to all slaves processor. For each K-means clustering iteration, new centroid locations are computed by master processor as well.

```
1:  MPI_Comm_rank (MPI_COMM_WORLD, &id);
2:  MPI_Comm_size (MPI_COMM_WORLD, &p);
3:  If(id==0):  Read in input file;
4:  If(id==0):  generate initial centroids location (C_j)^k ;
5:  If(id==0):  broadcast initial centroid location (MPI_Bcast(C_j ,0)) and distribute input
                data to other CPUs according to their id;
6:  do {
7:      for j = 1 to k
8:          (C_j ´)^k =0,   (count_j ´)^k =0;
9:      endfor;
10:     for i = id*(n/p) +1 to (id+1)*(n/p)
11:         for j = 1 to k
12:             compute distance_ij  (between X and centroid_j)
13:         end for;
14:         find the smallest distance between centroid_j and X;
15:         C_j ´+=X,  count_j ´ ++;
16:     end for;
17:     for j = 1 to k
18:         MPI_Reduce(C_j ´, C_j , MPI_SUM, 0);
19:         MPI_Reduce(count_j ´, count_j , MPI_SUM, 0);
20:         If(id==0): C_j /= (count_j );
21:     end for;
22:     MPI_Bcast(C_j ,0);
23: } until (Centroid does not move or some criteria is matched)
```

Figure 8.1 Parallel K-means algorithm Based on MPI

In order to prevent the bottleneck performance problem, we tried to minimize the computation for new centroid locations on master processor. We created one two-dimensional array (number of cluster × the dimensions of a single data) called Centroid array and one linear array (number of cluster) called Count Array for all processors. When a data point finds the belonging cluster i,

we add all data attribute values to ith row of Centroid array for each corresponding data dimension position and increment one the ith position of Count Array. In the end, master processor uses MPI_Reduce to obtain the summation value of both arrays and to calculate the new centroid locations by dividing every value on ith row of Centroid array by the value of ith position of Count Array. Figure 8.1 is the pseudo code for parallel K-means algorithm.

Step one through step five are the preparation steps for generating and broadcasting the initial centroids location and distributing the data points to each processors. Step nine to step sixteen is the assigning step for each data point finds their corresponding centroid. Step seventeen to twenty two work on recalculating centroids location. Although step seventeen to twenty looks like a bottleneck step, the computation and communication here is very small. Our time measurements ignore the I/O times, since we only care about the efficiency of the parallel K-means algorithm. Therefore, the time we record is between step5 to step23 in figure11.

8.3 Parallel Fuzzy C-means Clustering Algorithm

The same concept with both Centroid Array and Count Array can also be applied on FCM clustering algorithm. There are two major differences: for one, each data point has to compute the membership function for all clusters; also, the function for finding the new centroid locations is different. After each data point has computed their membership function, for each row in Centroid Array i, we add the multiplied value of all data attributes and $membership_i$ for each corresponding data dimension position. In stead of counting how many members belong to the cluster, Count Array works as a normalize value in Fuzzy C-means Clustering. After each data point has computed their membership function, for each cluster i, we add the $membership_i$ value

to ith position to Count Array. In the end, again, master processor uses MPI_Reduce to obtain the summation value of both array and calculate the new centroid locations by dividing every value on ith row of Centroid array by the value of ith position of Count Array.

8.4 FGK Parallelization Results

We implemented the parallelization algorithms on our FGK model. We ran all of our information granules in chapter 3 (detail information is listed in Table 3.1) on the Hydra machine with a maximum of 16 processors. We recorded the run time on all granules with 1, 2, 4, 8 and 16 CPUs. We also computed the speedup. Table 8.1 and 8.2 show the average execution time based on 5 independent iterations and the average speedup on all information granules for K-means clustering algorithm.

Table 8.1 Execution time on parallel K-means (in seconds) for all files

	File0	File1	File2	File3	File4	File5	File6	File7	File8	File9
Cpu=1	3666	940	1460	860	809	2853	3290	5.13	378	6.20
Cpu=2	1837	471	734	430	406	1428	2646	2.61	191	3.14
Cpu=4	919	238	368	216	203	716	829	1.35	95	1.62
Cpu=8	461	119	185	109	102	360	414	0.73	48	0.87
Cpu=16	232	60	93	55	52	181	209	0.43	24	0.50

Table 8.2 Speedup record on parallel K-means for all files

	File0	File1	File2	File3	File4	File5	File6	File7	File8	File9
Cpu=1	1	1	1	1	1	1	1	1	1	1
Cpu=2	1.997	1.997	1.988	1.998	1.998	1.997	1.998	1.966	1.981	1.975
Cpu=4	3.987	3.947	3.967	3.976	3.986	3.983	3.966	3.796	3.983	3.834
Cpu=8	7.939	7.900	7.859	7.889	7.919	7.920	7.939	7.037	7.861	7.123
Cpu=16	15.74	15.57	15.59	15.58	15.60	15.72	15.73	11.92	15.47	12.49

According to table 8.2, we see that besides the 2 smaller files (file7 and file9), all other files achieved linear speedup. Table 8.3 gives the average execution time and speedup for Fuzzy C-means Clustering Algorithm applied on our original data with ten cluster number. Due to more computation required by computing the new centroid locations, the speedup is not as good as K-means Clustering Algorithm. However, since the execution of FCM is required only once during the FGK model, the speedup of the whole model is still close to linear speedup. We calculate the final speedup of the FGK model, which executes one iteration of parallel FCM clustering algorithm and parallel K-means clustering algorithm, for five times. The final results are listed it in table 8.4 and are translated into Figure 8.2 for a clear visualization of linear speedup.

Table 8.3 Average execution time and Speedup on FCM with different number of processors

	Execution Time	Speedup
Cpu=1	19253.24	1
Cpu=2	9648.783	1.995
Cpu=4	4954.491	3.886
Cpu=8	2589.162	7.436
Cpu=16	1474.732	13.055

Table 8.4 Execution time and Speedup on FGK Model with different number of processors

	Execution Time	Speedup
Cpu=1	90589.89	1
Cpu=2	50392.53	1.797685
Cpu=4	22889.34	3.957733
Cpu=8	11587.16	7.818126
Cpu=16	6009.382	15.07474

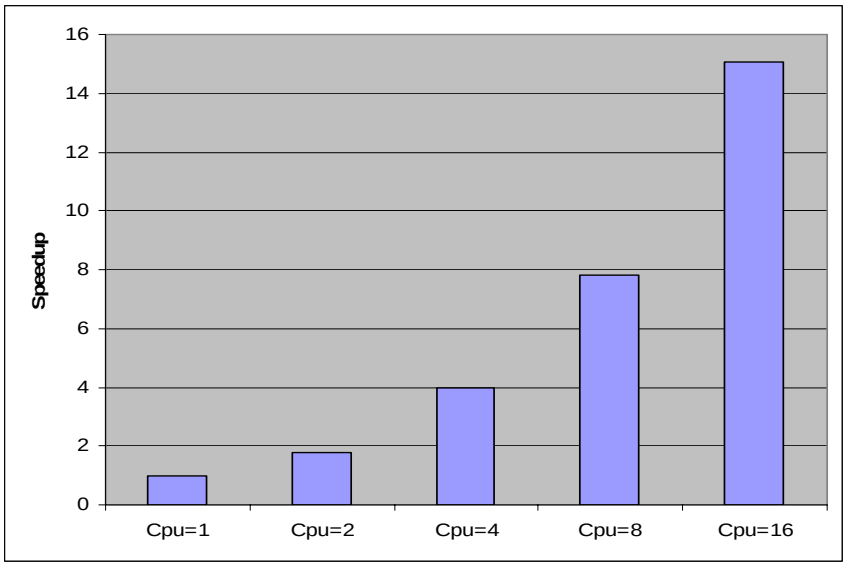


Figure 8.2 The relation between speedup and number of processors.

Compared to the performance between C and Python, the execution time of the program coded by C is much less then Python. Under the single CPU condition, unlike the program coded in Python which requires 17.88 days, the program written in C needs only 1.05 day. Since the hardware we used for the two programs is different (The hydra has 4G memory), the comparison may not be fair. However, we do save a lot of time. This facilitates us to find protein sequence motif information with different window sizes.

CHAPTER 9

SUMMARY AND FUTURE WORK

9.1 Summary

In this chapter, we focus on the analysis of the protein sequence recurring patterns. In the first two chapters, we give the fundamental biological information and explain how we obtain the experimental dataset. Next, we create and adapt two fuzzy granule computing models, Fuzzy Improved K-means model (FIK) and Fuzzy Greedy K-means model (FGK), onto the biological meaningful dataset to discovery protein sequence motifs. Due to the large dataset, we also performed high performance computing on the discovering process to dramatically decrease the computational cost. After that, we efficiently extract the motif information by the Super Granule Support Vector Machine Feature Elimination (Super GSVM-FE) model, which combines the power of the Rank-SVM and granule computing concepts. We justify the necessity of the extraction step and find the optimal tradeoff between the execution time and the obtained quality. Finally, we notice the problems caused by the fixed window size; therefore, we propose Super-Rule-Tree (SRT) structure constructed my our novel Hybrid Hierarchical K-means (HHK) clustering algorithm and a new association rule algorithm named Positional Association Rule algorithm to solve the problems. In the end, Positional Association Super-Rule algorithm is applied and yields admirable results. The flow of this dissertation can be easily understood by Figure 9.1.

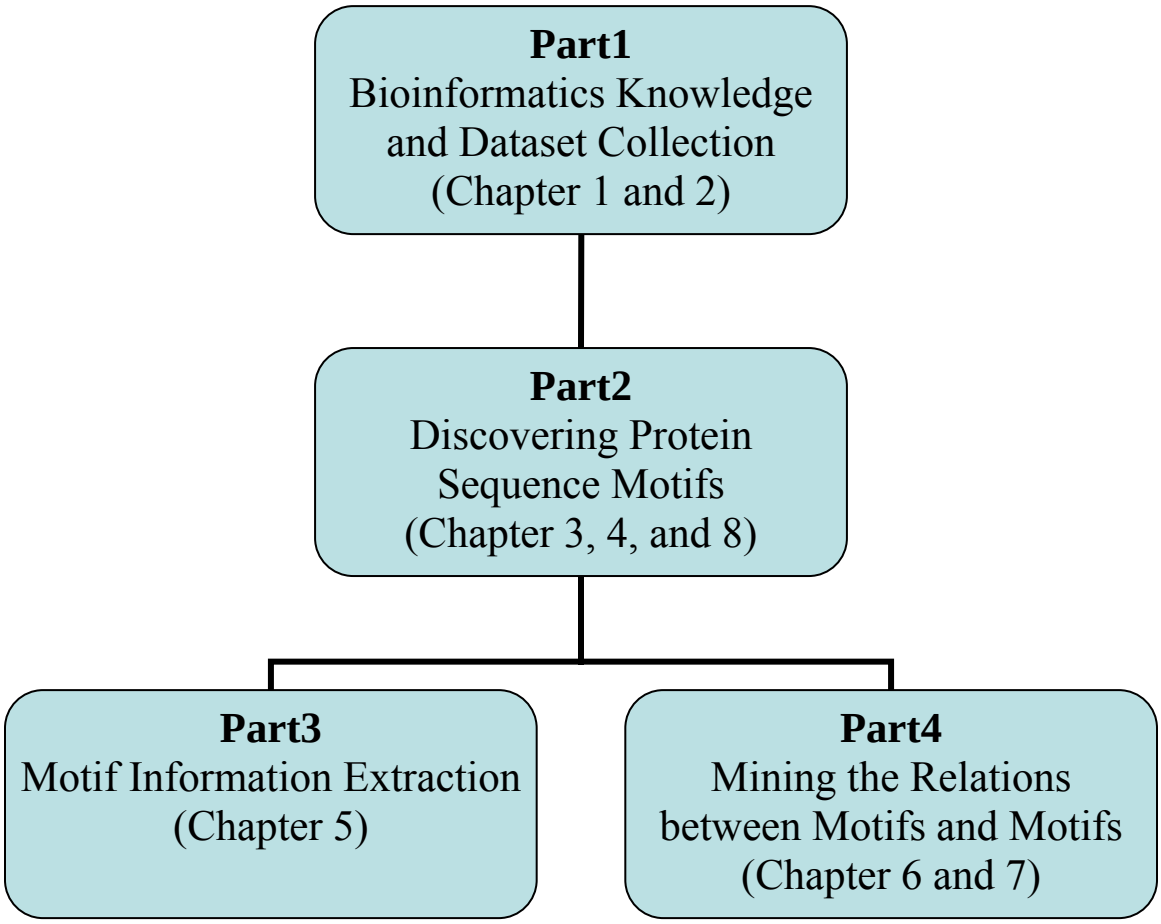


Figure 9.1 The summary of research flow in this dissertation

9.2 Achievements

Several achievements are made in each section. For the first part of the research, we collected 2710 different protein sequences as our initial input data source and generate more than 560,000 segments from it. This is one of the latest and largest dataset in the related field. An HSSP-BLOSUM62 measure is also proposed to evaluate the biochemical properties on the recurring patterns we found. HSSP profile and BLOSUM62 matrix are combined together, for the first time, to serve as an evaluation purpose.

For the second part of the work, we developed FIK model and FGK model. Each model contains a novel improved K-means clustering algorithm to smartly choose initial centroids. The powers of two granule computing models include (1) reduce time and space complexity, (2) filter outliers, and (3) generate better results. We have successfully reduced the execution to one-fifth and improved the quality of the protein sequence motifs. A new motif presentation format which combines motif logo was designed to give a more specific amino acid occurrence percentage. In order to further facilitate the step of protein sequence motif discovering, high performance computing also participated in the process. We have effectively achieved computing 15.07 times faster when 16 processors work together.

In the beginning of the third part of this work, we justified the necessity of feature elimination on our dataset by giving two major reasons: 1. the information we try to generate is about sequence motifs, but the original input data are derived from whole protein sequences by the sliding window technique. 2. During fuzzy c-means clustering, it has the ability to assign one segment to more than one information granule. However, not all data segments have direct relation to the granule assigned. After that, we proposed a novel granular feature elimination model called Super GSVM-FE, which combines Fuzzy C-means, Greedy K-means clustering algorithm and Ranking SVM, in order to extract protein sequence motif information. We also asserted a new research idea: while training SVM on clusters, it is not necessary to train all members in the cluster. By training only 80% of the members, we may obtain competitive results and reduce execution time dramatically.

We proposed several novel ideas and algorithms in the last part of this work. (1) Novel Hybrid Hierarchical K-means (HHK) clustering Algorithm, (2) Super-Rule-Tree (SRT) structure, (3) Positional Association Rule algorithm. More importantly, we merge all of these techniques

into Positional Association Super-Rule algorithm and applied in searching the relations between motifs and motifs. We also created a new HSSP-BLOSUM62 GAIN measure to locate the optimal parameter setting of Positional Association Super-Rules.

9.3 Future Work

Understanding the relation between protein sequence and their structure is one of the most important Bioinformatics research topics. Protein tertiary structure plays the most important role in determining the function of the protein. The biological methods to obtain protein tertiary structure are X-ray crystallography and Nuclear Magnetic Resonance (NMR). Both methods are very time consuming. Therefore, if we can avail ourselves of the close relationship between sequence and structural information, we may predict the protein local structural via protein sequence data. In this work, we put lots of effort into discovering and extracting protein sequence motifs. Based on these works, we obtained hundreds of very high quality motifs. We believe these motifs can play a crucial role in understanding the mysterious sequence-to-structure problem.

The power of our Super-GSVM-FE model is not limited to the research area of feature selection only. Since we have already performed Ranking-SVM on all clusters, when a new data segment comes in, we can use the trained models to identify which cluster the new segment belongs to based on its rank. Thus, if we include 3D structure in all of our clusters, when a new sequence segment comes in, we can have the power to predict its tertiary structure.

Although there's no direct linkage between part3 and part4 of our research, in Figure 9.1 the connection between two parts is necessary. The only reason we have not fed extracted motif

information into Positional Association Super-Rule algorithm is the limitation of time. Since the Ranking-SVM requires tremendous time to train (we have total 800 clusters, and the average number of members in each cluster is around 1000), we are waiting for the complete results while we continue developing Positional Association Rules. More than 500 high quality motifs can be expected after the process. When we apply all of those extracted motifs into our Positional Association Super-Rule algorithm, we look forward to constructing a higher level of SRT and finding more significant rules.

In conclusion, the major extension of our research can be viewed as Figure 9.2. Feed extracted motif information to Positional Association Super-Rule algorithm to make the linkage between part3 and part4. Add prediction mechanism to Super GSVM model to create a new research domain of our work. We believe the potential for additional progress in this dissertation is very strong.

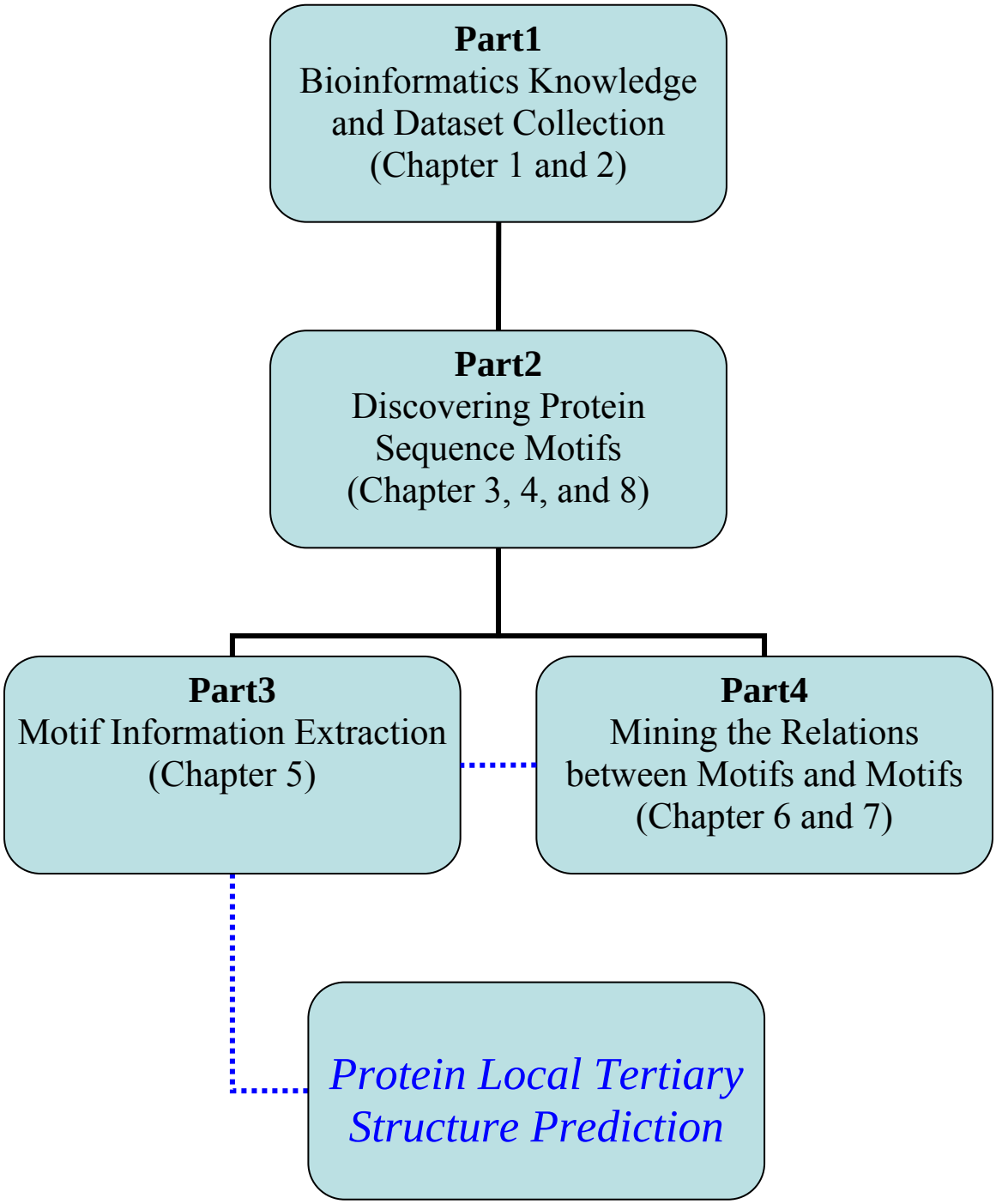


Figure 9.2 The summary of the future works

REFERENCES

1. Crick, F. (1970) Central dogma of molecular biology, *Nature. 227*, 561-3.

2. http://users.ugent.be/~avierstr/principles/centraldogma.html

3. Campbell, M. K. & Farrell, S. O. (2005) *Biochemistry*, Thomson Brooks/Cole.

4. http://en.wikipedia.org/wiki/Image:Protein-structure.png

5. Hulo, N., Sigrist, C. J. A., Le Saux, V., Langendijk-Genevaux, P. S., Bordoli, L., Gattiker, A., De Castro, E., Bucher, P. & Bairoch, A. (2004) Recent improvements to the PROSITE database, *Nucleic Acids Research. 32*, 134-137.

6. Attwood, T. K., Blythe, M. J., Flower, D. R., Gaulton, A., Mabey, J. E., Maudling, N., McGregor, L., Mitchell, A. L., Moulton, G. & Paine, K. (2002) PRINTS and PRINTS-S shed light on protein ancestry, *Nucleic Acids Research. 30*, 239-241.

7. Henikoff, S. (1999) Blocks+: a non-redundant database of protein alignment blocks derived from multiple compilations, *Bioinformatics. 15*, 471-479.

8. Zhong, W., Altun, G., Harrison, R., Tai, P. C. & Pan, Y. (2005) Improved K-means clustering algorithm for exploring local protein sequence motifs representing common structural property, *NanoBioscience, IEEE Transactions on. 4*, 255-265.

9. Bailey, T. L., Elkan, C., San Diego University of, C., Dept. of Computer, S. & Engineering. (1994) Fitting a Mixture Model by Expectation Maximization to Discover Motifs in Bipolymers.

10. Lawrence, C. E., Altschul, S. F., Boguski, M. S., Liu, J. S., Neuwald, A. F. & Wootton, J. C. (1993) Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment, *Science. 262*, 208-214.

11. Henikoff, S., Henikoff, J. G., Alford, W. J. & Pietrokovski, S. (1995) Automated construction and graphical presentation of protein blocks from unaligned sequences, *Gene. 163*.

12. Eskin, E. & Pevzner, P. A. (2002) Finding composite regulatory patterns in DNA sequences in pp. 354-363, Oxford Univ Press,

13. Price, A., Ramabhadran, S. & Pevzner, P. A. (2003) Finding subtle motifs by branching from sample strings in pp. 149-155, Oxford Univ Press,

14. Jensen, K. L., Styczynski, M. P., Rigoutsos, I. & Stephanopoulos, G. N. (2006) A generic motif discovery algorithm for sequential data, *Bioinformatics. 22*, 21-28.

15. Han, K. F. & Baker, D. (1995) Recurring Local Sequence Motifs in Proteins, *Journal of Molecular Biology. 251*, 176-187.

16. Wang, G. & Dunbrack, R. L. (2003) PISCES: a protein sequence culling server in pp. 1589-1591, Oxford Univ Press,

17. Sander, C. & Schneider, R. (1991) Database of homology-derived protein structures and the structural meaning of sequence alignment, *Proteins: Structure, Function & Genetics. 9*, 56-68.

18. Kabsch, W. & Sander, C. (1983) Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features, *Biopolymers. 22*, 2577-2637.

19. Davies, D. L. & Bouldin, D. W. (1979) A cluster separation measure, *IEEE Transactions on Pattern Analysis and Machine Intelligence. 1*, 224-227.

20. Henikoff, S. & Henikoff, J. G. (1992) Amino Acid Substitution Matrices from Protein Blocks, *Proceedings of the National Academy of Sciences of the United States of America. 89*, 10915-10919.

21. Lin, T. Y. (1999) Granular Computing: Fuzzy Logic and Rough Sets, *Computing with Words in Information/Intelligent Systems*, 183–200.

22. Lin, T. Y. (1998) Granular Computing on Binary Relations I: Data Mining and Neighborhood Systems, *Rough Sets in Knowledge Discovery. 1*, 107–121.

23. Lin, T. Y. (2000) Data Mining and Machine Oriented Modeling: A Granular Computing Approach, *Applied Intelligence. 13*, 113-124.

24. Yao, Y. Y. (2001) On Modeling data mining with granular computing, *Proceedings of the 25th Annual International Computer Software and Applications Conference*, 638-643.

25. Tang, Y., Zhang, Y. Q., Huang, Z. & Hu, X. (2005) Granular SVM-RFE gene selection algorithm for reliable prostate cancer classification on microarray expression data, *Bioinformatics and Bioengineering, 2005. BIBE 2005. Fifth IEEE Symposium on*, 290-293.

26. Yao, Y. Y. (2004) Granular computing, *Computer Science (Ji Suan Ji Ke Xue). 31*, 1-5.

27. Yao, Y. Y. (2004) A partition model of granular computing, *LNCS Transactions on Rough Sets. 1*, 232-253.

28. Yao, Y. (2005) Perspectives of granular computing, *Granular Computing, 2005 IEEE*

International Conference on. 1.

29. Yao, Y. Y. & Zhong, N. (2002) Granular computing using information tables, ***Data Mining, Rough Sets and Granular Computing*****, 102-124.**

30. Yao, Y. Y. & Yao, J. T. (2002) Granular computing as a basis for consistent classification problems, ***Proceedings of PAKDD. 2*****, 101–106.**

31. Lin, T. Y. Data Mining: Granular Computing Approach, ***Methodologies for Knowledge Discovery and Data Mining, Lecture Notes in Artificial Intelligence. 1574*****, 26-28.**

32. Han, J. & Kamber, M. (2006) ***Data Mining: Concepts and Techniques*****, Morgan Kaufmann.**

33. Dunn, J. C. (1973) A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, ***Journal of Cybernetics. 3*****, 32-57.**

34. Bezdek, J. C. Pattern Recognition with Fuzzy Objective Function Algorithms. 1981 in, Plenum Press, New York,

35. http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/cmeans.html#dunn

36. Chen, B., Tai, P. C., Harrison, R. & Pan, Y. (2006) FIK Model: Novel Efficient Granular Computing Model for Protein Sequence Motifs and Structure Information Discovery, ***BioInformatics and BioEngineering, 2006. BIBE 2006. Sixth IEEE Symposium on*****, 20-26.**

37. Chen, B., Tai, P. C., Harrison, R. & Pan, Y. (2006) FGK Model: An Efficient Granular Computing Model for Protein Sequence Motifs Information Discovery, ***IASTED proc. International conference on Computational and Systems Biology (CASB), Dallas*****.**

38. Crooks, G. E., Hon, G., Chandonia, J. M. & Brenner, S. E. (2004) WebLogo: A Sequence Logo Generator, ***Genome Research. 14*****, 1188-1190.**

39. Ohler, U. & Niemann, H. (2001) Identification and analysis of eukaryotic promoters: recent computational approaches, ***Trends in Genetics. 17*****, 56-60.**

40. He, J., Chen, B., Hu, H. J., Harrison, R., Tai, P. C., Dong, Y. & Pan, Y. (2005) Rule Clustering and Super-rule Generation for Transmembrane Segments Prediction, ***IEEE Computational Systems Bioinformatics Conference Workshops (CSBW'05)*****, 224-227.**
41. Jain, A. K. & Dubes, R. C. (1988) ***Algorithms for clustering data*****, Prentice-Hall, Inc. Upper Saddle River, NJ, USA.**

42. MacQueen, J. (1967) Some methods for classification and analysis of multivariate observations, ***Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and***

***Probability. 1*, 14.**

43. Hu, J., Ray, B. K. & Singh, M. (2007) Statistical methods for automated generation of service engagement staffing plans-References, *IBM Journal of Research and Development 51*, 281-293.

44. Bradley, P. S. & Fayyad, U. M. (1998) Refining Initial Points for K-Means Clustering, *Proc. 15th International Conf. on Machine Learning. 727*.

45. Brown, D. E. & Huntley, C. L. (1990) A Practical Application of Simulated Annealing to Clustering.

46. Pelleg, D. & Moore, A. (2000) X-means: Extending K-means with efficient estimation of the number of clusters, *Proceedings of the 17th International Conf. on Machine Learning*, 727–734.

47. Zhang, T., Ramakrishnan, R. & Livny, M. (1996) BIRCH: an efficient data clustering method for very large databases, *Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, 103-114.

48. Karypis, G., Han, E., Kumar, V. & Minnesota Univ Minneapolis Dept Of Computer, S. (1999) *Multilevel Refinement for Hierarchical Clustering*, Defense Technical Information Center.

49. Chen, B., Tai, P. C. & Harrison, R. (2005) Novel hybrid hierarchical-K-means clustering method (HK-means) for microarray analysis, *Computational Systems Bioinformatics Conference, 2005. Workshops and Poster Abstracts. IEEE*, 105-108.

50. Garey, M. R. & Johnson, D. S. (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness*, WH Freeman & Co. New York, NY, USA.

51. Agrawal, R., Imielinski, T. & Swami, A. (1993) Mining association rules between sets of items in large databases, *ACM SIGMOD Record. 22*, 207-216.
52. Park, J. S., Chen, M. S. & Yu, P. S. (1995) An effective hash-based algorithm for mining association rules, *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, 175-186.

53. Agrawal, R. & Srikant, R. (1994) Fast algorithms for mining association rules, *Proc. 20th Int. Conf. Very Large Data Bases, VLDB. 1215*, 487499.

54. Han, J. & Fu, Y. (1999) Mining multiple-level association rules in large databases, *Knowledge and Data Engineering, IEEE Transactions on. 11*, 798-805.

55. Kuok, C. M., Fu, A. & Wong, M. H. (1998) Mining fuzzy association rules in databases, *ACM SIGMOD Record. 27*, 41-46.

56. Srikant, R. & Agrawal, R. (1996) Mining Sequential Patterns: Generalizations and Performance Improvements, *Advances in Database Technology--EDBT'96: 5th International Conference on Extending Database Technology, Avignon, France, March 25-29, 1996: Proceedings*.

57. Zaki, M. J. (2000) Sequence mining in categorical domains: incorporating constraints, *Proceedings of the ninth international conference on Information and knowledge management*, 422-429.

58. Srikant, R. & Agrawal, R. (1997) Mining generalized association rules, *FUTURE GENER COMPUT SYST. 13*, 161-180.

59. Han, J. & Fu, Y. (1995) Discovery of multiple-level association rules from large databases, *Proceedings of the 21th International Conference on Very Large Data Bases*, 420-431.

60. Toivonen, H. (1996) Sampling large databases for association rules, *Proceedings of the 22th International Conference on Very Large Data Bases*, 134-145.

61. Liu, B., Hsu, W. & Ma, Y. (1998) Integrating classification and association rule mining, *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, 80-86.

62. Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H. & Verkamo, A. I. (1994) Finding interesting rules from large sets of discovered association rules, *Proceedings of the third international conference on Information and knowledge management*, 401-407.

63. Brin, S., Motwani, R. & Silverstein, C. (1997) Beyond market baskets: generalizing association rules to correlations, *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, 265-276.

64. Pasquier, N., Bastide, Y., Taouil, R. & Lakhal, L. (1999) Discovering frequent closed itemsets for association rules, *Lecture Notes in Computer Science. 1540*, 398-416.

65. Srikant, R., Vu, Q. & Agrawal, R. (1997) Mining association rules with item constraints, *KDD. 97*, 67–73.

66. Agrawal, R. & Shafer, J. C. (1996) Parallel Mining of Association Rules.

67. Zaki, M. J., Parthasarathy, S., Ogihara, M. & Li, W. (1997) New algorithms for fast discovery of association rules, *3rd Intl. Conf. on Knowledge Discovery and Data Mining. 20*.

68. Fukuda, T., Morimoto, Y., Morishita, S. & Tokuyama, T. (1996) Data mining using two-dimensional optimized association rules: scheme, algorithms, and visualization, *Proceedings of the 1996 ACM SIGMOD international conference on Management of data*,

13-23.

69. Han, E. H. S., Karypis, G. & Kumar, V. (2000) Scalable Parallel Data Mining for Association Rules, *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 337-352.

70. Piatetsky-Shapiro, G., Frawley, W. J., Brin, S., Motwani, R., Ullman, J. D., Aggarwal, C. C., Yu, P. S., Liu, B. & Hsu, W. (2005) Discovery, Analysis, and Presentation of Strong Rules}, *Proceedings of the 11th international symposium on Applied Stochastic Models and Data Analysis ASMDA. 16*, 191-200.

71. Lent, B., Swami, A. & Widom, J. (1997) Clustering association rules, *Proceedings of the Thirteenth International Conference on Data Engineering*, 220-231.

72. Ozden, B., Ramaswamy, S. & Silberschatz, A. (1998) Cyclic association rules, *Data Engineering, 1998. Proceedings., 14th International Conference on*, 412-421.

73. Fukuda, T., Morimoto, Y., Morishita, S. & Tokuyama, T. (1999) Mining Optimized Association Rules for Numeric Attributes, *Journal of Computer and System Sciences. 58*, 1-12.

74. Cheung, D. W., Han, J., Ng, V. T., Fu, A. W. & Fu, Y. (1996) A fast distributed algorithm for mining association rules, *Parallel and Distributed Information Systems, 1996., Fourth International Conference on*, 31-42.

75. Cai, C. H., Fu, A. W. C., Cheng, C. H. & Kwong, W. W. (1998) Mining association rules with weighted items, *Database Engineering and Applications Symposium, 1998. Proceedings. IDEAS'98. International*, 68-77.

76. Wu, X., Zhang, C. & Zhang, S. (2002) Mining Both Positive and Negative Association Rules, *Proceedings of the Nineteenth International Conference on Machine Learning table of contents*, 658-665.

77. Icev, A., Ruiz, C. & Ryder, E. F. (2003) Distance-Enhanced Association Rules for Gene Expression, *3rd ACM SIGKDD workshop on Data Mining in Bioinformatics*, 34-40.

78. Kam, H. J., Lee, D. & Lee, K. H. (2003) Mining and interpretation of association rules among protein sequence motifs, *Engineering in Medicine and Biology Society. Proceedings of the 25th Annual International Conference of the IEEE*, 3551-3554.

79. Zar, J. H. Biostatistical analysis. 1999, *Upper Saddle River, NJ*.

APPENDIX

Related works:

Journal Publications:

1. **Bernard Chen**, Stephen Pellicer, Phang C. Tai, Robert Harrison and Yi Pan, "Efficient Super Granular SVM Feature Elimination (Super GSVM-FE) Model for Protein Sequence Motif Information Extraction", International Journal of Functional Informatics and Personalised Medicine, to appear.

Conference Publications:

1. **Bernard Chen**, Stephen Pellicer, Phang C. Tai, Robert Harrison and Yi Pan, "Super Granular Shrink-SVM Feature Elimination (Super GS-SVM-FE) Model for Protein Sequence Motif Information Extraction", ***IEEE BIBE 2007,Boston***, Accepted. (**First Runner-Up Best Student Research Work**)

2. **Bernard Chen**, Stephen Pellicer, Phang C. Tai, Robert Harrison and Yi Pan, "Super Granular SVM Feature Elimination (Super GSVM-FE) Model for Protein Sequence Motif Information Extraction", ***IEEE CIBCB 2007, Hawaii***, proceeding pp.317-323.

3. Xuezheng Fu, **Bernard Chen**, Yi Pan and Robert Harrison. "Statistical Estimate for the Size of the Protein Structural Vocabulary" ***ISBRA 2007, Altanta***, proceeding pp.530-538

4. Bernard Chen, Phang C. Tai, Robert Harrison, and Yi Pan, "FGK model: A Efficient Granular Computing Model for Protein Sequence Motifs Information Discovery", ***IASTED CASB 2006, Dallas***, proceeding pp56-61.

5. Bernard Chen, Phang C. Tai, Robert Harrison, and Yi Pan, "FIK model: A Novel Efficient Granular Computing Model for Protein Sequence Motifs and Structure Information Discovery", ***IEEE BIBE 2006, Washington D.C. ,*** proceding, pp20-26.

6. Bernard Chen, Phang C. Tai, Robert Harrison, and Yi Pan, "Novel Clustering Algorithm Combined With DSSP Post Processing For Protein Sequence Motif Discovering", ***IEEE GrC 2006, Atlanta,*** proceeding, pp.449-452

7. B. Chen, P. Tai, R. Harrison, and Y. Pan, "Novel Hybrid Hierarchical-Kmeans Clustering (H-K-means) for Microarray analysis," ***IEEE CSB2005, Stanford,*** Workshops and Poster Abstracts, pp.105-108

8. J. He, **B. Chen**, H Hu, R. Harrison, P. Tai, Y. Dong, and Y. Pan, "Rule Clustering and Super-Rule Generation for Transmembrane segments prediction," ***IEEE CSB2005, Stanford,*** workshops and Poster Abstracts, pp. 224-227

Book Chapters:

1. J. He, H. Hu, **B. Chen**, H Hu, R. Harrison, P. Tai, Y. Dong, and Y. Pan, "Rule Extraction from SVM for Protein Structure Prediction", **Rule Extraction form Support Vector Machines,** Chapter 10, pp. 227-252