Spring 5-5-2012

Innovative Algorithms and Evaluation Methods for Biological Motif Finding

Wooyoung Kim

INNOVATIVE ALGORITHMS AND EVALUATION METHODS FOR BIOLOGICAL MOTIF FINDING

by

WOOYOUNG KIM

Under the Direction of Dr. Yi Pan

ABSTRACT

Biological motifs are defined as overly recurring sub-patterns in biological systems. Sequence motifs and network motifs are the examples of biological motifs. Due to the wide range of applications, many algorithms and computational tools have been developed for efficient search for biological motifs. Therefore, there are more computationally derived motifs than experimentally validated motifs, and how to validate the biological significance of the 'candidate motifs' becomes an important question. Some of sequence motifs are verified by their structural similarities or their functional roles in DNA or protein sequences, and stored in databases. However, biological role of network motifs is still invalidated and currently no databases exist for this purpose.

In this thesis, we focus not only on the computational efficiency but also on the biological meanings of the motifs. We provide an efficient way to incorporate biological information with clustering analysis methods: For example, a sparse nonnegative matrix factorization (SNMF) method is used with Chou-Fasman parameters for the protein motif finding. Biological network motifs are searched by various clustering algorithms with Gene ontology (GO) information. Experimen-

tal results show that the algorithms perform better than existing algorithms by producing a larger number of high-quality of biological motifs.

In addition, we apply biological network motifs for the discovery of essential proteins. Essential proteins are defined as a minimum set of proteins which are vital for development to a fertile adult and in a cellular life in an organism. We design a new centrality algorithm with biological network motifs, named MCGO, and score proteins in a protein-protein interaction (PPI) network to find essential proteins. MCGO is also combined with other centrality measures to predict essential proteins using machine learning techniques.

We have three contributions to the study of biological motifs through this thesis; 1) Clustering analysis is efficiently used in this work and biological information is easily integrated with the analysis; 2) We focus more on the biological meanings of motifs by adding biological knowledge in the algorithms and by suggesting biologically related evaluation methods. 3) Biological network motifs are successfully applied to a practical application of prediction of essential proteins.

INDEX WORDS: Biological network motif, Clustering analysis, Gene ontology, Essential protein, Machine learning

INNOVATIVE ALGORITHMS AND EVALUATION METHODS FOR BIOLOGICAL MOTIF FINDING

by

WOOYOUNG KIM

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

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2012

INNOVATIVE ALGORITHMS AND EVALUATION METHODS FOR BIOLOGICAL MOTIF FINDING

by

WOOYOUNG KIM

Committee Chair: Dr. Yi Pan

Committee: Dr. Raj Sunderraman

Dr. Alex Zelikovsky

Dr. Dr. Yichuan Zhao

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

May 2012

ACKNOWLEDGEMENTS

This dissertation would have not been possible without supports of many persons. First of all, I am greatly thankful to my advisor, Prof. Yi Pan for providing a challenging and exciting topic of this dissertation, but also for his supervise and encouragement. I also want to express my profound acknowledgements to Dr. Min Li and Dr. Jianxin Wang who gave excellent advise and guided my work to the right direction. It is also a pleasure to thank Prof. Haesun Park for her great support on the start of the research with her novel algorithm of sparse nonnegative matrix factorization. Special thanks also go to Dr. Bernard Chen and Mr. Jingu Kim for their helps on various aspects including data collection and design of algorithms.

I also would like to thank my committee, Prof. Raj Sunderraman, Prof. Alex Zelikovsky and Prof. Yichuan Zhao for the valuable suggestions and all the supports.

Last, but not least, I thank my family; my parents, mother-in-law, husband and my precious two girls for their love and all the supports. Especially, I am deeply grateful for my husband's infinite patience that accompanied me along this long journey.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . . . **iv**

LIST OF TABLES . . . . . **ix**

LIST OF FIGURES . . . . . **xiii**

LIST OF ABBREVIATIONS . . . . . **xxi**

Chapter 1 INTRODUCTION **1**

1.1 Motivation 1

1.2 The Approaches 4

1.3 Contribution 6

1.4 Dissertation Road Map 7

Chapter 2 LITERATURE REVIEW **8**

2.1 Biological Motif 8

2.1.1 Sequence Motif 9

2.1.2 Network Motif 17

2.2 Biological Network 30

2.2.1 Types of Biological Networks 30

2.2.2 Network Property 35

2.2.3 Challenges 36

2.3 Granular Computing 36

2.3.1 Data Clustering Algorithms 36

2.3.2 Network Clustering 40

Chapter 3 LARGE-SCALE OF PROTEIN MOTIF DISCOVERY WITH SNMF AND CHOU-FASMAN PARAMETERS **49**

3.1 Background 49
3.2 Problem Statement 50
3.3 Methods 51
3.3.1 Related Works 52
3.3.2 New Approaches 53
3.3.3 Evaluation Methods 57
3.4 Result and Discussion 58
3.4.1 Data set and data representation 59
3.4.2 Experiment steps 60
3.4.3 Clustering Results 60
3.4.4 Sequence Motifs 63
3.5 Summary and Future Work 63

Chapter 4 ALGORITHMS AND EVALUATION METHODS FOR BIOLOGICAL NETWORK MOTIF 70

4.1 Background 70
4.2 Problem Statement 72
4.3 Methods 72
4.3.1 Definitions 73
4.3.2 Algorithms 74
4.3.3 Evaluation Methods 83
4.4 Result and Discussion 85
4.5 Summary and Future Work 90

Chapter 5 ESSENTIAL PROTEIN DISCOVERY IN A PPI NETWORK USING NETWORK MOTIF AND GENE ONTOLOGY 95

5.1 Background 95
5.2 Problem Statement 96
5.3 Methods 97

5.3.1 Algorithms . . . 97
5.3.2 Evaluation Methods . . . 103
5.4 Results and Discussion . . . 106
5.4.1 Experimental data . . . 106
5.4.2 Evaluation by three evaluation measures . . . 107
5.4.3 Effects of EDGEGO . . . 108
5.4.4 Analysis of MC and MCGO . . . 109
5.5 Summary and Future Work . . . 110

Chapter 6 MODEL-DRIVEN APPROACH TO PREDICTING ESSENTIAL PROTEINS IN A PPI NETWORK . . . 117

6.1 Background . . . 117
6.2 Problem Statement . . . 118
6.3 Methods . . . 119
6.3.1 Algorithms . . . 119
6.3.2 Evaluation Methods . . . 123
6.4 Results and Discussion . . . 126
6.4.1 Data sets and features . . . 126
6.4.2 Comparison of the balanced data sets . . . 126
6.4.3 CENT-GO and CENT-ING-GO . . . 128
6.4.4 Prediction based on different classifiers . . . 129
6.4.5 Analysis on each centrality measure . . . 131
6.5 Summary and Future Works . . . 133

Chapter 7 CONCLUSIONS . . . 139

7.1 Conclusions . . . 139
7.2 Future Work . . . 142

REFERENCES . . . 144

APPENDICES **178**

Appendix A NMF AND BMF **178**

A.1 Nonnegative Matrix Factorization 178

A.2 Sparse Nonnegative Matrix Factorization 180

A.3 Bounded Matrix Factorization 181

A.3.1 Two Block Coordinate Descent Framework for BMF 182

A.3.2 BLS based on the Active Set method 183

A.3.3 Fast Combinatorial Bounded Least Squares 186

Appendix B PARALLEL NETWORK MOTIF SEARCH **191**

B.1 Parallel search of network motifs 191

B.1.1 Recover Subgraphs from Removed Edges 191

B.2 Network clustering and parallel search 193

LIST OF TABLES

Table 2.1: Number of Non-isomorphic Subgraphs for undirected and directed graphs with up to 10 vertices [1] 22

Table 3.1: Chou-Fasman parameter 68

Table 3.2: Comparison of various clustering methods 69

Table 4.1: Various algorithms used for the detection of biological network motifs: All the algorithms introduced in this work are compared based on the type, the time before enumeration by ESU, parameter and its deterministic property. Here d is GO depth threshold, l is the number of GO terms associated to the graph G, c is the number of clusters, r is the number of edges to remove, and η, β for sparse NMF computation. 83

Table 4.2: Results of 4-node biological network motifs in the *DIP Core*network: We can see that EDGEBETWEENNESS-BNM performs best in 'motif included in complex' measure while EDGEGO-BNM performs best in other measures. 87

Table 4.3: Results of 5-node biological network motifs in the *DIP Core* network: We can see that EDGEBETWEENNESS-BNM performs best in 'motif included in complex' measure while EDGEGO-BNM performs best in other measures. 87

Table 4.4: Results of 4-node biological network motifs in the *Y2k* network: We can see that EDGEBETWEENNESS-BNM performs best in 'motif included in complex' measure. NMFGO-BNM performs best on 'MF' and 'CC clustering score' measures. EDGEGO-BNM performs best in the 'motif included in functional module' measure 'BP, CC clustering score' measures. However all the algorithms perform poorly in 'MF clustering score' measure, with less than 30. 88

Table 4.5: Results of 5-node biological network motifs in the *Y2k* network: We can see that EDGEBETWEENNESS-BNM performs best in 'motif included in complex' measure while EDGEGO-BNM performs best in other measures. 89

Table 4.6: Y2k statistical properties, from FANMOD: Each type of 4-node subgraph shows its significance based on its structural uniqueness. The label is generated by *Nauty* program [2] and the corresponding shape is shown in Figure 4.3. In this network, the first three types are detected as network motifs. 89

Table 4.7: Y2k reduced network by EDGEGO-BNM statistical properties, from FANMOD: Each type of 4-node subgraph shows its significance based on its structural uniqueness. The label is generated by *Nauty* program [2] and the corresponding shape is shown in Figure 4.3. In this network, the first three types are detected as network motifs. 91

Table 4.8: The rates of motifs included in a 'rRNA processing' functional module in the yeast (Y2k network), computed using Equation (5.18): Except ESU, all algorithms search 30% of subgraphs in the original network. However, EDGEGO-BNM recovers over 90% of motifs included in functional module. We note that the non-motif types of Cr, CF and CR have a number of instances for this functional match, indicating structural uniqueness is insufficient to discover its biological significance. 91

Table 5.1: Confusion matrix or contingency table 112

Table 5.2: Area under curve (AUC) value for each PR curve 112

Table 6.1: Comparison of balanced data sets: To verify that all the 10 data sets are statistically similar, we run a meta classifier to each data set and obtain an AUC-ROC value. We verified that all the data sets are statistically similar through Mann-Whitney U-statistics test with their AUC-ROC values. . 127

Table 6.2: Statistical Significant of the set of integrated features: Each set of features was assessed based on its statistical significance. We can observe that when we run a classifier to the set of integrated features (CENT-ING-GO), the performance improves significantly. 130

Table 6.3: The performances of CENT-GO, ING-GO and the combined of the two, CENT-ING-GO are compared with their area under ROC (AUC-ROC), area under PR (AUC-PR), accuracy (ACC) and time (T). CENT-GO and ING-GO have slight variations, but integration of them, CENT-ING-GO, can improve the performance significantly. 131

Table 6.4: Comparison of classifiers: The CENT-ING-GO feature set is performed with 2 different classifiers and the performances are compared based on AUC-ROC, AUC-PR, ACC and T measures. "Classifier by Kim et al., 2012" performs better than "Classifier by Acencio and Lemke, 2009." . 132

Table 6.5: Statistical Significance of each feature: Each feature was assessed based on its statistical significance. The performance improvement of CENT-GO is statistically verified as all the p-values, compared with each measure, are less than 0.05. 135

Table 6.6: Comparison of each measure: The experiment with CENT-GO is compared with the experiment with a single feature each. The performances are compared based on its AUC-ROC, AUC-PR, ACC and T measures. 136

LIST OF FIGURES

Figure 2.1: An example of position weight matrix (PWM) representation for a sequence motif of length six. Profiles are variations of PWM. 11

Figure 2.2: An example of profile HMMs, from the paper [3]. A DNA motif at the top can be converted into a profile HMM at the bottom. 12

Figure 2.3: An example of a sequence logo, from the paper [4]. 13

Figure 2.4: Top-scoring pathway alignments between bacteria and yeast. (a) The protein-protein interaction networks of *H. pylori* and *S. cerevisiae* were globally aligned to reveal conserved network regions through b-f processes. This figure is from [5]. 19

Figure 2.5: Outline of the Graemlin algorithm, by the courtesy of [6]. (A) Four networks with their phylogenetic relationships. (B) Graemlin first performs a pairwise alignment of the two closest species, using d-cluster and a pair of seeds. (C) Graemlin extends the seed using a greedy algorithm. (D) Graemlin transforms the resulting alignment and the unaligned nodes into three generalized networks for use in the next step. (E) In the next step, Graemlin will perform three pairwise alignments, one for each of the newly created generalized network. 42

Figure 2.6: A query network g and a target network G are given in the left-side hand. The resulting graph G' is constructed in the right hand side, using Path-Match algorithm. Dashed lines show vertex correspondences and $\equiv$ in G' means the representing vertex in G. In this example, at most one mismatches or indels are allowed between two matches. Figure is from the paper [7]. 43

Figure 2.7: Subgraph search time increases rapidly as the motif size increases. The horizontal axis is the size of motifs and the vertical axis is the time consumed for the search. The dashed line is an exponential curve to show a trend of search time based on the size of motifs. 43

Figure 2.8: Subgraph search time increases rapidly as the size of a network increases. The horizontal axis is the number of edges and the vertical axis is the time consumed for the search. The dashed line is a polynomial curve to show a trend of search time based on the size of a network. 44

Figure 2.9: The emergent integrated gene regulation network representing the cell progress in a mammalian cell. The signaling pathway has begun to lay out a circuitry that will likely mimic electronic integrated circuits in complexity and finesse. Gene expression process has much overlap regions with signaling pathways. The figure is from [8]. 44

Figure 2.10: Overview of signaling pathways in the baker's yeast *S. cerevisiae*. The activated receptor activates intracellular processes. The figure is from [9]. 45

Figure 2.11: Transcription regulatory network in yeast. The figure is from [10]. . . . 46

Figure 2.12: Example view for a metabolic network. The figure is from [11]. 47

Figure 2.13: An example view of a protein-protein interaction network 48

Figure 2.14: A taxonomy of clustering approaches, by the courtesy of [12]. 48

Figure 3.1: The top image is the coefficient matrix when $k = 3$ and bottom image is the coefficient matrix when $k = 45$. The y-axis represents the number of clusters and the x-axis is the data point. For a specific data shown as a red vertical box, the assignment of the top matrix is clearer than the bottom matrix, as the second row clearly beats the others. The bottom coefficient matrix has more than 7 non-zero values holding around 10% of the weight each, making a proper assignment difficult. 55

Figure 3.2: A number of protein sequences in a protein family obtained from PDB server are aligned on the left. According to the frequencies of twenty amino acids represented as one-letter codes, the proteins are expressed as a profile data on the right figure. Sliding a window of length 9, the 20×9 matrix shown inside the red box represents one protein segment data format. 60

Figure 3.3: This figure summarizes the experiment steps in this study. The original data set of a primary sequence is divided into smaller subsets (information granules) with double applications of FCM. For each subset, secondary structure information is inferred with Chou-Fasman parameters and added to each data set. SNMF is finally applied to each subset and the result is evaluated using two evaluation criteria, secondary structure similarity and sDBI. 61

Figure 3.4: Helices motif with conserved A 64

Figure 3.5: Helix-Turn motif 65

Figure 3.6: Turn-Sheet motif 66

Figure 3.7: Sheet-Turn motif 67

Figure 3.8: Helix-Turn-Helix motif 67

Figure 4.1: After modifying the graph: Original network (left) and the modified network (right) after removing edges or clustering the graph. As shown in the right hand side, a number of clusters and a list of removed edges are provided as a result. 75

Figure 4.2: An example of GO graph view (GO DAG), where the root node is depth 0. If a GO is depth 0, then it is the most general term, meaning most of genes or proteins are annotated with this GO term. As the depth of the GO increases, the information of GO gets specific. 77

Figure 4.3: Shapes and labels for 4-node subgraphs in an undirected network: There are six types for 4-node subgraphs in an undirected network. Each type is labeled by *Nauty* program. 90

Figure 4.4: DIP Core network: Search ratios based on the subgraph type: The ratio of frequency of each type is relatively preserved and it indicates that our algorithms can be used for the structural network motif discovery as well. Relative frequencies of each algorithm is plotted with different colors of line. The horizontal axis indicates each subgraph type for 4-node subgraphs. The vertical axis shows the relative frequency of each type. The values are shown in the table below the figure. 92

Figure 4.5: Y2k network: Search ratios based on the subgraph type: The ratio of frequency of each type is relatively preserved and it indicates that our algorithms can be used for the structural network motif discovery as well. The description of the plots and the table is same as in Figure 4.4. 93

Figure 5.1: The top graph is an original network. If we remove A, then the graph is separated into two subgraphs as shown in the bottom-left side. However, if we remove B or C, the graph is nearly scattered as appeared in the bottom-right side. Therefore, a central node is not deterministic. The graph is captured from [13]. 99

Figure 5.2: TR proportion: Each bar indicates the performance result of DC, BC, CC, SC, EC, SoECC, LAC, MC and MCGO from the left to right. 108

Figure 5.3: Statistical measures including SN, SP, PPV, NPV, F and ACC: Each bar indicates the performance result of DC, BC, CC, SC, EC, SoECC, LAC, MC and MCGO from the left to right. 109

Figure 5.4: PR curves: MCGO is at the most upper-right-hand side indicating as the best algorithm. 110

Figure 5.5: TR proportion: Each bar indicates the performance result of DC, DCGO, SoECC, SoECCGO, LAC, and LACGO from the left. 111

Figure 5.6: Statistical measures: Each bar indicates the performance result of DC, DCGO, SoECC, SoECCGO, LAC, and LACGO from the left. 112

Figure 5.7: PR curves: Each -*GO* algorithm is better than its original algorithm. . . 113

Figure 5.8: TR proportion: Each bar indicates the performance result of BC, BCGO, CC, CCGO, SC, SCGO, EC and ECGO from the left. 113

Figure 5.9: Statistical measures: Each bar indicates the performance result of BC, BCGO, CC, CCGO, SC, SCGO, EC and ECGO from the left. 114

Figure 5.10: PR curves: Each -*GO* algorithm is better than its original algorithm. . . 114

Figure 5.11: TR proportion: Each bar indicates DCGO, BCGO, CCGO, SCGO, ECGO, SoECCGO, LACGO and MCGO from the left. 115

Figure 5.12: Statistical measures: Each bar indicates DCGO, BCGO, CCGO, SCGO, ECGO, SoECCGO, LACGO and MCGO from the left. 115

Figure 5.13: PR curves: The curve of MCGO is at the most upper-right-hand side. . 116

Figure 5.14: The graphical view generated from the MINT web site [14] for ALG1 and their neighbor nodes. (b) shows the extended nodes which are neighbors of WBP1 in (a). 116

Figure 6.1: The process of CENT-GO extraction based on centrality measures from GO-pruned PPI network: In the left, an yeast PPI network is pruned with EDGEGO algorithm, where 14,925 interactions are removed out of 37,209 interactions total. For each vertex, eight centrality measures are calculated from the GO-pruned PPI, each of which is a feature of the protein node. The imbalanced data set is under-sampled to form a balanced data set. . 122

Figure 6.2: The process of ING-GO extraction based on an integrated network and BP GO and CC GO terms: An yeast PPI, transcriptional regulatory and metabolic network is integrated into an integrated (INGI) network. 12 topological features are extracted from this INGI and 11 features are obtained from biological process and cellular localization GO terms. Each protein consists of 23 features and a balanced data set is also obtained with undersampling. 123

Figure 6.3: *Classifier by Kim et al, 2012*: 7 decision-tree based algorithms, a support vector machine (SVM) and neural network method, to each of which 'bagging' is applied for variance reduce, are combined into a meta-classifier. 124

Figure 6.4: *Classifier by Acencio and Lemke, 2009* : 8 decision-tree based algorithms to each of which 'bagging' is applied for variance reduce, are combined into a meta-classifier. 125

Figure 6.5: ROC curves of the ten balanced data sets with CENT-ING-GO features: All 10 data sets have similar performances. 128

Figure 6.6: ROC curves for individual feature sets and CENT-GO: The prediction with CENT-GO performs significantly better than with each individual measure. We also notice that DCGO, MCGO, LACGO, and SoECCGO shows relatively good scores which are characterized as local features. 129

Figure 6.7: PR curves for individual feature sets and CENT-GO: The prediction with CENT-GO performs significantly better than with each individual measure. We also notice that DCGO, MCGO, LACGO, and SoECCGO show relatively good scores which are characterized as local features. 130

Figure 6.8: ROC curves of ING-GO (Kim et al, 2012), CENT-GO (Acencio and Lemke, 2009) and the CENT-ING-GO (CENT-GO + ING-GO): The prediction performance improves significantly with the integral of the two sets of features. 131

Figure 6.9: PR curves of ING-GO (Kim et al, 2012), CENT-GO (Acencio and Lemke, 2009) and the CENT-ING-GO (CENT-GO + ING-GO): The prediction performance improves significantly with the integral of the two sets of features. 132

Figure 6.10: ROC curves for the two classifiers are provided. Classifier by Kim et al., 2012 is slightly better than Classifier by Acencio and Lemke, 2009. . . 133

Figure 6.11: PR curves for the two classifiers are provided. Classifier by Kim et al., 2012 is slightly better than Classifier by Acencio and Lemke, 2009. . . 134

Figure 6.12: Decision tree on the balanced dataset5 with CENT-GO features with 64 instances per leaf: The data set contains only CENT-GO features and the tree algorithm generate a rule where "MCGO" as a root. The values are normalized before running the algorithm, and it produces 72% of accuracy and the area under ROC is $.734$. The eclipses are the features and in this set, "MCGO" and "ECGO" are likely to determine the essentiality of proteins. 137

Figure 6.13: Decision tree on the balanced dataset5 with combined features of CENT-ING-GO with 64 instances per leaf: The data set contains 31 features and the tree algorithm generate a rule where "MCGO" as a root. The values are normalized before running the algorithm, and it produces73% of accuracy and the area under ROC is $.752$. The eclipses are the features and in this set, "MCGO" and "ECGO", "clustering coefficient (c)", "nucleus" and "endoplasmic reticulum (er)" are likely to determine the essentiality of proteins. 138

Figure B.1: An example network $G = (V, E)$ with $|V| = 16, |E| = 19$. This is an original network. 195

Figure B.2: After applying a clustering algorithm to the original network of Figure B.1. Four edges are removed as a result. 196

Figure B.3: The process of recovering missing subgraphs from removed edges of Figure B.2 196

Figure B.4: After clustering, some clusters are isomorphic. For example, we obtain three clusters with this type of subgraph after clustering. 197

Figure B.5: 3-node subgraphs are enumerated using ESU [15] algorithm. 197

LIST OF ABBREVIATIONS

- NMF - Nonnegative Matrix Factorization
- BMF - Bounded Matrix Factorization
- SNMF - Sparse NMF
- PPI - Protein-Protein Interaction
- GO - Gene Ontology
- BNM - Biological Network Motif
- TRN - Transcriptional Regulatory Network
- BP - Biological Process
- MF - Molecular Function
- CC - Cellular Component
- DC - Degree Centrality
- BC - Betweenness Centrality
- CC- Closeness Centrality
- SC- Subgraph Centrality
- EC- Eigenvector Centrality
- SoECC- Sum of Edge Clustering Coefficient
- LAC-Local Average Connectivity
- MC- Motif Centrality

- ROC- Receiver Operating Characteristic
- AUC- Area Under Curve
- PR - Precision Recall
- ACC- Accuracy
- TP - True positive
- FP - False positive
- TN - True negative
- FN - False negative
- SVM- Support Vector Machine
- NN- Neural Network
- NNLS - Nonnegative Least Square
- BLS - Bounded Least Square
- LSI - Least Squares with Inequality constraints
- LSE - Least Square Equality
- CSSLS - Combinatorial Subspace Least Squares
- MPI - Message Passing Interface
- LRE - List of Removed Edges
- RSRE - Recover Subgraphs from Removed Edges
- PLE - Previously Removed Edges

Chapter 1

INTRODUCTION

1.1 Motivation

Biological motifs include sequence motifs and network motifs. Sequence motifs are short substrings in DNA or in proteins that seem to occur more often than usual. On the other hand, network motifs are subgraph patterns that occur frequently and uniquely in a network. Based on this definition, network motifs are not limited to biological networks only, but can be applied to any other networks. Network motifs are used to appreciate the structures of networks locally and, surprisingly, in most cases the networks seem to be largely composed of these small connected network motifs. Uri Alon et al. [16] used network motifs to describe functional building blocks in a gene regulation network, followed by many applications of network motifs in biological networks.

In proteins, sequence motifs are discovered when a collection of diverse proteins share a common function or structure in a few common residues. If the proteins are enzymes, the motifs are involved in the chemical catalysis in the active site. Meanwhile, biological significance of network motifs in a protein-protein interaction (PPI) networks is unclear yet. Biological functions of network motifs with small size have been studied mostly in gene regulation networks. Gene regulation networks control the gene expression in response to biological signals in the cell, therefore network motifs are defined as patterns of gene regulation. However, usages of network motifs in PPI are limited to a few applications such as the relationship to evolutionary conservation or the prediction of protein interactions. Still, the use of network motifs in PPI is focused more on their structural properties than on biological functions.

Network motifs are developed to describe local properties of a network, with the advent of other local computing algorithms including network clustering, network alignment and network querying. The resulting sub-networks by network clustering can reveal specific local properties in the network as well as save computing times in a great amount. Traditional clustering algorithms

have been developed to deal with enormous data to discover a number of 'intrinsic' similarities in clusters and 'hidden' difference between the clusters. Network clustering is an unsupervised classification as there is no prior classification criteria, and an unsupervised non-predictive learning as there is no trained characterization. Network alignment and network querying are also related with network motifs, but their processes involve testing of biological similarities unlike network motifs. Network motifs are unique among these local analyses in some extents as they are defined by their topological properties only. As the search involves high computational challenges, computational efficiency has the highest priority in most of motif search algorithms. There were exact counting algorithms and approximation search algorithms.

Exhaustive recursive search (ERS) [16], enumerate subgraphs (ESU) [15] and compact topological motifs [17] are exact counting algorithms. However, because of the high computing demands for exact counting, several approximation algorithms have been provided including search based sampling (MFinder) [16]), randomized ESU (Rand-ESU) [18]) and NeMoFinder [19]. Although these approximation algorithms are feasible, false detection is highly possible. Therefore, parallel algorithms have been developed for feasible exact counting of network motifs [20, 21].

There are many biological applications of network motifs as well. Network motifs were initially introduced as functional building blocks in transcriptional regulatory networks [16, 22]. Distinct network motifs have provided information about typical patterns in different types of biological networks. Przulj et al. [23] used network motifs as a relative graphlet frequency distance to compare various protein-protein interaction networks. Also motif frequencies are exploited as classifiers for network model selection [24]. Milo et al. [25] studied that networks of different biological and technological domains have been classified into different superfamilies on the bases of motif significance profiles. Albert et al. [26] used network motifs successfully to predict protein interactions. Network motifs are closely related to evolutionary conservation as well. In the study by Conant and Wagner [27], network motifs in transcriptional regulatory networks are not evolutionary conserved while network motifs in PPI networks are evolutionary related. From this work, it is concluded that groups of proteins are more evolutionary conserved than individual protein. On the other hand, network motifs are extended to 'motif modes' which has a certain topology plus a

specific functional property. The number of motif modes in the study [28] reaches up to a million, which differentiated various evolutionary constraints. The motifs founded by LaMoFinder algorithm [29] are similar to motif modes in a sense that they have topological and biological properties as well. In LaMoFinder, network motifs are labeled according to the border informative functional class (FC) of gene ontology (GO) terms. Border informative FC-GO terms are GO terms which have at least 30 directly annotated proteins and their parents GO terms have less than 30 annotated proteins. The labeling process involves a clustering task with informative FC-GO terms being features for each network motif.

Through a number of network motif algorithms and applications in biological networks, however, we notice several problems. First, biological meanings of network motifs are not validated thoroughly. Network motifs are selected only by their structural uniqueness and only small part of motif instances are examined and utilized to some applications. We believe that biologically-related evaluation methods should be provided as we are studying biological networks. Next, it is possible that we waste most of computational time to count the instances of a particular network motif which will be thrown away later. We should have a pre-filtering task which will help for efficient search of biologically important motif instances. Also, non-network motifs (that is, structurally insignificant subgraphs) have not been analyzed in any studies, which are filtered out before applied to any applications. It is likely that we lose many biologically meaningful subgraphs by ignoring non-network motifs. Another problem is that other than the traditional notion of network motifs being functional building blocks, there may be other aspects that network motifs can represent. Besides, network motifs bear numerous issues such as an optimal size of network motifs and additional knowledge for effective discovery, which will be potential research topics in the future.

This thesis is, therefore, dedicated to solve those problems in network motifs and, more generally, in biological motifs. For protein motifs, our analysis targets to obtain universally preserved sequence patterns across protein family boundaries. We utilize clustering algorithms to analyze the whole data set and evaluate its biological importance with their structural similarities. For network motifs, we seek biological meanings of network motifs, with innovative algorithms and evaluation methods which are designed to define biologically significant network motifs, defined

as **Biological Network Motifs**. Biological network motifs are biologically significant small-size of subgraphs regardless of its structure. These might not be exactly categorized into a number of different classes, but some biological roles can be assigned to them. We emphasize that our work is a preliminary work toward comprehensive researches on network motifs focusing on their biological usages and construction of network motif databases. We introduce a number of algorithms for efficient searches of as many biological network motifs as possible, and design new evaluation methods, which validate the biological quality of network motifs. Furthermore, we exploit biological network motifs to the application of predicting essential proteins in a PPI network.

1.2 The Approaches

Biological motifs are mostly discovered through computational approaches, where the main challenges occur by the followings reasons. First, there is no prior knowledge of how the motifs look or how large they are. Second, the location of motifs is also unknown. Therefore, exact search for biological motifs usually takes exponential time. In addition, the insertion or deletion in the sequences alignment makes it more difficult to find sequence motifs. Although there is no deletion nor insertion, detection of network motifs has more computing challenges since the process requires three-dimensional search, isomorphic testing which is NP-hard problem, and repeated processes for uniqueness determination.

With all the above problems, however, we focus more on biological meanings of motifs. Unlike traditional sequence motif search which is to find a consensus short-substring from a set of functionally related sequences, we search protein sequence motifs which can describe universally preserved sequence patterns across protein family boundaries. For efficient and better qualified result, we make use of a clustering method, especially nonnegative matrix factorization. We also show that an incorporation of Chou-Fasman parameter, which is a statistical information for protein secondary structure of each amino acid, helps further improvements on the discovery of protein motifs. The new algorithm is compared with an improved K-means algorithm by Zhong et al. [30], FIK by Chen et al. [31] and FGK by Chen et al. [32], based on the secondary structure similarity

(SSS) and structural DBI (sDBI) measure. In fact, the sDBI measure is developed in this work, to qualify the clustering results with more weight on the structural properties.

For network motifs, we provide the following approaches;

- We define biological network motifs that emphasize more on biological significance rather than topological significance.
- We introduce a number of algorithms for an efficient search of biological network motifs using clustering analysis and additional biological information.
- Although biological functions of network motifs are not fully appreciated until now, we design some evaluation measures that can measure biological values of network motifs with limited sources.
- To see if biological network motifs can be practically applied to many applications, we apply biological network motifs to essential protein discovery and prediction problems, and provide experimental results.

The algorithms compete with existing algorithms and the performances are compared based on the new measurements introduced in this work. The main strategy of efficient search is to modify the original network by reducing a number of edges: We provide edge-removing algorithms and clustering algorithms. After the modification, all algorithms produce a number of clusters and a number of edges between clusters. They are computationally efficient algorithms because the number of biological network motif search reduces in a great amount with the decrease of the number of edges as shown in Figure 2.8. Through experiments with a couple of *S. cerevisiae* PPI networks, we demonstrate that we can save the search time polynomially while preserving the detection rate for different patterns. This explains that the algorithms can be used to traditional network motif discovery as well.

To compare the performances, we develop three evaluation methods: motifs included in protein complex, motifs included in functional module and GO Term clustering score. Experimental results show that our algorithms beat existing algorithms of ESU, Rand-ESU [15] and

MFinder [33]. Furthermore, we can parallelize the whole process using message passing interface (MPI) as we obtain a number of disjoint sub-networks as the result of these algorithms. In addition to the motif search in each sub-network, we can search the missing subgraphs from the removed edges if we use an algorithm RSRE (Recover Subgraphs Form Removed Edges) at Appendix B, and RSRE can be trivially parallelized as well.

Biological network motifs are applied to discover essential proteins in a PPI network. Many existing centrality algorithms have been used to detect essential proteins and their performances were compared in a number of studies. In this thesis, we develop a Motif Centrality with GO (MCGO) [34] which uses network motifs for a more robust centrality algorithm and incorporates biological information of gene ontology (GO) terms. We show that MCGO performs best in among other centrality algorithms for the detection of essential proteins. Also, various biological centrality algorithms, where biological pruning process preceded, are integrated to form a set of features to be plugged into a machine learning algorithm to predict essential proteins. Previous study by Acencio and Lemke [35] extracted a number of features from an integrated biological networks, BP (biological process) and CC (cellular component) GO annotations. This feature set is referred as 'ING-GO' (Acencio and Lemke, 2009) in this work. We show that our set of features with biological centrality algorithms, named 'CENT-GO' (Kim et al., 2012), includes much less number of features, but produces the almost same performance as that of 'ING-GO' (Acencio and Lemke, 2009). Also we improve the prediction rate significantly by integrating 'CENT-GO' (Kim et al. 2012) and 'ING-GO'(Acencio and Lemke 2009).

1.3 Contribution

In this thesis, we utilize clustering algorithm for efficient biological motifs discovery including protein sequence motifs and network motifs. However, to obtain biologically meaningful results, we have to involve biological information in the algorithms and evaluate the results with biological standards.

Overall, the work has three contributions to the study of biological motifs: 1) We used clustering methods to reveal the properties of motifs in an efficient way, and to involve biological

information in the process. In fact, biological clusters and biological motifs are closely related. As we do not have prior knowledge in motifs, clustering the data with intrinsic similarity helps efficient discovery of motifs. 2) We raised various questions regarding biological motif applications and we specifically designed algorithms and evaluation methods based on the questions. We designed a number of algorithms which combine the topological and biological information of biological data. Since most of the algorithms are based on biological and topological information, the results are more consistent than existing algorithms which are based on random selections. We also provided a number of evaluation measures which qualify biological importance of the biological motifs. As we know of, this is the first attempt to suggest systematical evaluation measurements for network motifs. 3) We show that biological network motifs are successfully applied to a practical application, which is the prediction of essential proteins in a network. With these contributions, we hope that our work gives a guideline for the researches in biological motifs.

1.4 Dissertation Road Map

The remaining part is organized as the followings. We review related literatures about biological motifs, biological networks and granular computing in Chapter 2. Protein sequence motifs in a large-scale data set are discovered using clustering method combined with biological information in Chapter 3. Definition of biological network motif, introduction of new algorithms and evaluation methods are explained in chapter 4. We apply biological network motifs to identify or predict essential proteins in Chapter 5 and Chapter 6, followed by a conclusion and future study in Chapter 7. The detail of NMF algorithm with the Bounded Matrix Factorization (BMF) which is a generalized NMF, and the parallel search setup for network motifs are presented in the Appendix.

Chapter 2

LITERATURE REVIEW

We provide background information for biological motifs in this chapter. We review biological motifs such as sequence motifs and network motifs in Section 2.1, followed by biological networks in Section 2.2. As biological data, in general, is huge, we utilize granular computing strategy for efficient computation, which will be discussed in the following section. Through this review, we will bring up some of issues regarding biological motifs and their meanings and provide solutions in next chapters.

2.1 Biological Motif

Biological motifs are defined as recurring patterns in biological systems and they are presumed to have biologically important structures or functions. In DNA or RNA sequences, motifs are nucleotide patterns that appear most frequently in a set of DNA or RNA sequences. They imply sequence-specific binding sites for transcription factor proteins (TF) [36], or relate significant RNA processes such as ribosome binding and transcription termination [37].

Protein sequence motifs, consisting of twenty amino-acids, have different definitions and interpretations according to Bork and Koonin [38]; 1) They are short functional motifs which are independently evolved from the surroundings. Examples include myristilation sites and glycosylation sites. 2) Some involve short structural motifs which are repeating super-secondary structures. 3) Others are functional motifs which do not involve invariant residues, rather involve sequence level constraints, including transmembrane regions, signal sequences or cell sorting. 4) They are discovered from a set of protein sequences with homology tests and reflect functional and structural constraints from the given set. In the past they are discovered through biological and chemical experiments, but more and more of them are being discovered from computational methods these

days. In this thesis, we discuss only the last type of motifs which involve the concept of homology and consensus.

Similar to sequence motifs, network motifs are frequent and unique patterns but discovered from networks instead of sequences. While sequence motifs were first derived from applications such as discovery of DNA binding sites or core functional subsequence in proteins, network motifs were first introduced as structurally significant patterns in a network and a number of applications are followed afterwards. Therefore, most of algorithms focus on finding structurally and statistically significant patterns, but the biological meanings of the results are discussed only through some applications. In fact, detecting network motifs requires high computational resources which limits measuring the quality of network motifs in biological aspects.

For better understanding of the problems for biological motifs and the challenges, in this section, we review biological motifs including sequence motifs and network motifs, introduce some computational challenges for search, and examine the evaluation measures with various aspects.

2.1.1 Sequence Motif

Researches on sequence motifs were motivated from the discovery of DNA binding sites. DNA binding sites are short subsequences in DNA which are bound by DNA-binding proteins. The problem consists of two subproblems [39]: 1) Given a collection of known binding sites, find a representation for prediction in a newly discovered sequence; 2) Given a set of sequences containing binding sites for a common transcription factor, find the location and the representation. The representation is expressed as a subsequence in the set, which is later defined as a sequence motif. Through many technologies for DNA sequencing [40, 41], the amount of DNA sequences to be analyzed has increased rapidly. Therefore, in parallel with experimental approaches such as DNase footprinting, gel-shift or reporter construct assays [37], many computational algorithms have been developed to discover DNA binding sites, and they are generalized for the task of discovery of sequence motifs not only in DNA but also in RNA and protein sequences.

Protein sequence motifs are defined similarly, as particular amino acid sequences which are characteristic of a specific biochemical function. One example of protein motif is zinc finger motif.

DNA motifs and protein motifs can be related each other. For example, it is generally known that if a sequence motif is detected in the exon of a gene, it is related to the structural motif of a protein. If it is not in the exon, it can be regulatory sequence motif. Nevertheless, the problems for DNA and protein motifs are generalized as sequence motif problems as the differences are mostly on the bases in the sequence. Sequence motifs have various representations and different algorithms for detection or prediction. Most algorithms are limited to specific type of representation, and a lot of tools have been developed for detection. We review the diversity of motif representations and many computational algorithms in this section.

2.1.1.1 Representation

Sequence patterns have been described in various ways [42], in order to summarize the gathered information, usually after multiple sequence alignments. Since sequence motifs are short patterns of sequences, the representations also follow these existing sequence representations. Ferreira and Azevedo [43] categorized those representations as deterministic and probabilistic representations [44] and we will discuss them within this category.

Consensus sequence representation is an example of a deterministic representation. It is the result of multiple sequence alignment and refers to the most common elements, which are nucleotide or amino acid, at a specific location. PROSITE database [45] stores sequence motifs with consensus sequence representation, where regular expression syntax is used for searching sequence motifs represented in consensus sequence. Regular expression is a notational algebra describing a string or a set of strings, and the rules are as the followings, details of which are also described in [42].

- The standard one-letter codes for nucleotide or amino acids are used.
- ‘x’ is used for an arbitrary element.
- Multiple letters for one location are listed in square parentheses, [ ] .
- The elements, which are not allowed, are listed in $\{\}$.

- To separate the elements, '-' is used.
- If an element is repeated, it is specified with a numerical value in parenthesis, (). For example, x(3) means, x-x-x. x(1,3) means x, or x-x or x-x-x.
- For a pattern restricted to either N- or C-terminal, it starts with '<', or ends with '>'.
- The pattern ends with a period.

For example, if a PROSITE pattern is "A-x-[ST](2) - x(0,1) - V-LI", it means "Ala-any-[Ser or Thr]-[Ser or Thr] - (any or none) - Val - (any but Leu, Ile)". The deterministic representation is, however, too rigid to represent the diversity of motifs in most cases. In addition, as motifs are very short, the simple representation can lead many false positives in databases. Therefore, a number of stochastic representations are introduced, including Position Weight Matrix(PWM), Profiles or profile HMMs.

PWM, also called as a Position Specific Scoring Matrix (PSSM) or Position Specific Weight Matrix (PSWM), is one of probabilistic motif representations. A PWM is constructed from multiple alignments of sequences and provide a weighted score representing the variation in each column. A profile is a variation of PWM, but while PWM does not allow gaps, profile includes gap penalties in the alignments. Figure 2.1 is an example of PWM, and profiles are in a similar format.

A	-38	19	1	12	10	-48
C	-15	-38	-8	-10	-3	-32
G	-13	-48	-6	-7	-10	-40
T	17	-32	8	-9	-6	19

Figure 2.1: An example of position weight matrix (PWM) representation for a sequence motif of length six. Profiles are variations of PWM.

Profiles can be represented as profile HMMs where the alignment results are represented as hidden Markov models. Profile HMMs is another variation of PWM, but it is appropriate for searching databases for remotely homologous sequences [46]. Profile HMMs representation was first introduced in computational biology field by Churchill [47], which are used as profile models by Krogh [3] later. An example of conversion from a DNA motif into a profile HMM representation is shown in Figure 2.2. The conversion steps will not be described in this thesis, so readers are advised to refer [3] for more details.

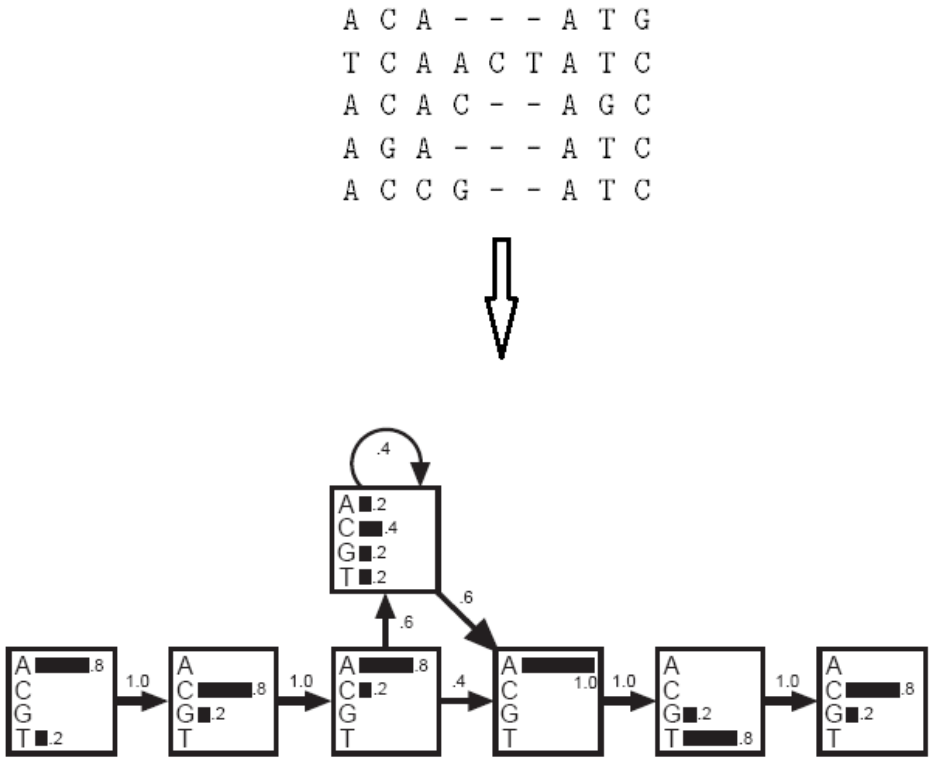


Figure 2.2: An example of profile HMMs, from the paper [3]. A DNA motif at the top can be converted into a profile HMM at the bottom.

On the other hand, a sequence logo [4] is a graphical description of a profile, where the size of a symbol indicates the frequency that a given base appears at the specific position in the sequence. The letters at each location are sorted so that the most common element is on the top. Sequence logos help determine the consensus sequence as well as the relative frequency of elements and the

information content at each position, with graphical advantages. One example is shown in Figure 2.3. A web tool, or a WebLogo [48] is available to generate a sequence logo at the following site [$http://weblogo.threeplusone.com$].

Figure 2.3: An example of a sequence logo, from the paper [4].

2.1.1.2 Algorithms

Sequence motifs, which had been discovered through biological and chemical experiments, are now detected and predicted through various computational methods which are based on sequence alignment algorithms. Most algorithms are computationally very expensive with the following unknown factors: 1) what the motifs look like including the size and the composition, and 2) where they start.

Although, currently, more than a hundred publications of algorithms exist, there is no comprehensive benchmark to compare the algorithms. Only some surveys of the algorithms are available. D'haeseleer [49] categorized the motif search algorithms into three approaches: enumeration, deterministic optimization and probabilistic optimization. In this thesis, we follow this category to explain the existing algorithms and describe some of tools that implemented the algorithms.

Enumeration Enumeration algorithms exhaustively search for the location with all possible candidate 'n-mers' sub-sequences. Such algorithms try to cover the entire search space so that

avoid poor local optimum. Dictionary-based methods search all the possible cases in the target set of sequences, while block-based methods search the space up to a given length (YMF). WeeberWeb [50] allows some mismatches and uses an efficient suffix tree representation to find motifs. Enumeration algorithms such as WeederWeb and YMF perform especially well on eukaryotic sequence sets of known motifs in the study [51]. But as they are computationally too expensive, only a small data set is feasible to be applied. The tools include MITRA [52], Weeder [53] and YMF [54].

Deterministic Optimization Deterministic optimization algorithms simultaneously optimize a motif, especially PWM format, through iteration steps. As one of deterministic optimization method, expectation-maximization (EM) has been used for the PWM data sets [37, 55]. In the algorithm, a PWM for a candidate motif is initialized, then in the expectation step (E-step), the probability that it was generated by the motif is computed in each data. Then we take a weighted average across the probabilities to refine a new motif model in the maximization step (M-step). The EM-steps are iterated until it converges to a maximum of the log likelihood. MEME [56] is a variation of EM and performs one iteration of EM for each n-mer subsequence from the data set, then selects the best motif to iterates only with the selected one, which avoids a poor local optimal point. Improbizer [57], MEME [55] are available tools which are based on these algorithms.

Probabilistic Optimization While deterministic optimization algorithms take a weighed average across all n-mers, probabilistic optimization algorithms take a weighted sample from the n-mers. Gibbs sampling algorithm [58] is the example. Gibbs sampling algorithm, as a stochastic implementation of EM, initializes a motif with a number of randomly selected subsets. Then all the subsequences in the data set is scored based on the initial model. Through iterations, the model is refined by adding or removing a new subsequence, and the binding probabilities are updated. Available tools of this method include AlignACE [59], GLAM [60], MotifSampler [61] and SeSiMCMC.

The judge about which algorithm is better highly depends on the settings, such as motif representation, objective function and the number of data. Therefore, it is advised to combine the results from multiple motif finding tools and decide the biological relevant with an appropriate evaluation criterion.

2.1.1.3 Evaluation

Determining which algorithm to use for a specific application is practically important but nontrivial as almost every algorithm uses a different measure to optimize or score motifs. An important issue on motifs is that not all of the resulting motifs are useful and majority of them arise by chance [62]. Consequently, evaluation measures to determine significant motifs have been another major trends. The most obvious way of assessing the significance of the motifs is to delegate the decision to the biologists or chemists. However, it is unrealistic as there are vast amount of motifs to be evaluated. Alternatively, automatic evaluation measures have been introduced with statistical and informative measures, although these methods do not guarantee the biological significance [63]. We review some of automatic evaluation methods in this section.

For automatic evaluation, the problem tends to be restricted into a classification problem where a set of positive and negative patterns are available. In general, a significance measure is introduced [64] as a function $f(m, C) \rightarrow \mathbb{R}$, where m is the motif being investigated and C is a set of background sequences or target family. The return value is a score of the m based on C. The target motif m is compared under the C. Therefore, C will be a positive set and the remaining set $\overline{C}$ is a negative set. Then the motifs are evaluated according to 1) the probabilities of matching a random sequence, 2) sensitivity and specificity, 3) information content and 4) minimum description length, as Sagot suggested in [65].

The measures are also referred as class-based, theoretic-information or mixed measures in [62]. Class-based measures include sensitivity, specificity, positive predictive value, F-measure and discrimination power (Dp) [66]. Theoretic-information measures evaluate the degree of information in a motif. Information gain(IG) [67, 68], minimum description length (MDL) [69, 70], log-odds (L) [3] and Z-score [62] measures are the examples of theoretic-information measures.

J-measure [71] and mutual information (I–measure) [72,73] measures are provided as combination measures of class-based and theoretic-information measures.

D'haeseleer provides more evaluation measures in his paper [49], such as, information content [37], log likelihood and MAP score which are based on statistical models, to see how much a motif deviates from a background distribution. Additional methods, including group specificity (site specificity) [74], sequence specificity [50,51], and positional bias [74,75] or uniformity [51], try to distinguish real motifs from spurious motifs.

However, we should note that different measures have different properties. Although these measures are commonly used to evaluate motif search algorithms, none of them is sufficient to distinguish the real motifs from false ones [51]. Hence, the best measure should be chosen based on its applications. Ferreira and Azevedo [62] compared the evaluation measures with PROSITE patterns and showed their relevances; some are closely related and others are very exclusive. For other measures, Zhong et al. [76] developed a secondary structural similarity measures, Chen et al. [32] proposed a HSSP-BLOOSUM62 measure to impose the chemical property of motifs and Kim et al [77] introduced a structural DBI (sDBI) to emphasize the structural quality of motifs.

2.1.1.4 Applications

Other than the DNA binding sites, DNA sequence motifs are useful in defining genetic regulatory networks, deciphering the regulatory program of individual genes or predicting regulatory networks [37]. Protein sequence motifs also have many applications other than supporting proteins' structure and functional information. For example, protein sequence motifs helped to discover sub-families in large protein families [69]. In addition, many tasks for the family classification used protein sequence motifs [78–84].

Regardless of the sources, however, sequence motifs are very useful to sequencing analysis and the applications as well. For example, sequence motifs help to perform clustering task in [85]. Sequence clustering problem is usually challenged by the lack of efficient similarity measure, therefore, sequence motifs were used in the similarity measure. In [86], sequence motifs are used

for the sequence annotation or for the gene expression analysis. In addition, they are very useful to detect any homology relations between sequences or larger structures.

2.1.2 Network Motif

Network motif is a subgraph pattern which appears more than usual in a network. Although it is obvious that the definition of network motifs is derived from the concept of sequence motif, the detection of network motifs does not involve alignment, instead it involves isomorphic testing, which is NP-hard problem, and statistical evaluations such as Z-score or P-value for uniqueness determination, which requires vast amount of repetitions. Therefore, most researches have focused on fast revelation of network motifs as the process involves computationally challenging steps. Identified network motifs are applied to many real-world problems including protein function prediction, detecting evolutionary conservation and specific genes and so on. In this research, we present a number of algorithms to find network motifs and many applications of network motifs. We address some of issues in network motif and raise the need for biological network motif and its systematic evaluation methods.

2.1.2.1 Network Motif and Beyond

Network motif is a repeated subgraph pattern in a network and it is identified by only its topological frequency and uniqueness [16, 87, 88]. On the other hand, network alignment and network querying, which are similar to network motif, use both topological and biological information. In this section, we first compare network motifs with sequence motifs, then review network alignment and network querying. Network alignment and network querying is developed in the context of biological networks, which will be covered in Section 2.2.

Sequence Motif and Network Motif A sequence motif is a repeated pattern that is prevalent in a number of sequences, such as DNA/RNA or protein sequences. Sequence motifs are known to have biological significance such as binding sites and conserved domains. If a motif is in the exon of a gene, it can encode a *structural motif* which is a three dimensional motif determining

a unique element of the overall structure of a protein. With this property, sequence motifs can predict other proteins' structural or functional behaviors. Therefore, discovering sequence motifs is a key task to comprehend the connection of sequences with their structures. PROSITE [45], PRINTS [89] and BLOCKS [90, 91] are currently the most popular motif databases. There are also many software programs for discovering one or more candidate motifs from a number of nucleotide or protein sequences. These include PhyloGibbs [92], CisModule [93], WeederH [94], and MEME [56]. For example, MEME utilizes hidden Markov models (HMM) to generate statistical information for each candidate motif.

Network motif is defined as similar to a sequence motif, except it can be discovered in networks. Network motif is a subgraph which appears more than usual in a network and it is identified with its topological uniqueness. While sequence motifs allow some variations in multiple alignment such as gabs or indels, network motifs allow no deviations in structures, and no alignment is required. However, discovery of network motifs requires much higher computing resources as it involves isomorphic testing and a number of random generations to determine its statistical significance. Furthermore, the biological roles of network motifs are still unclear whereas a number of sequence motifs are proven to have clear biological functions and many motif databases categorize the motifs based on their different roles.

Network Alignment Network alignment, along with network querying, belongs to biological network comparison tasks. Network alignment requires both a scoring function and a search procedure in the same way as sequence alignment. PATHBLAST for pairwise alignment is first developed [5] and it is generalized to multiple alignments in Graemlin [6]. In [5], two protein-protein interaction networks are aligned using interaction topology and protein sequence similarity, which identified conserved interaction pathways and complexes. Figure 2.4, by the courtesy of [5], shows the processes and results.

Network alignment first merges the two networks according to protein sequence similarity then connections are established based on the graph match. Like sequence alignment, network alignment has 'match', 'gab' and 'mismatch' concepts for scoring function. The pairwise net-

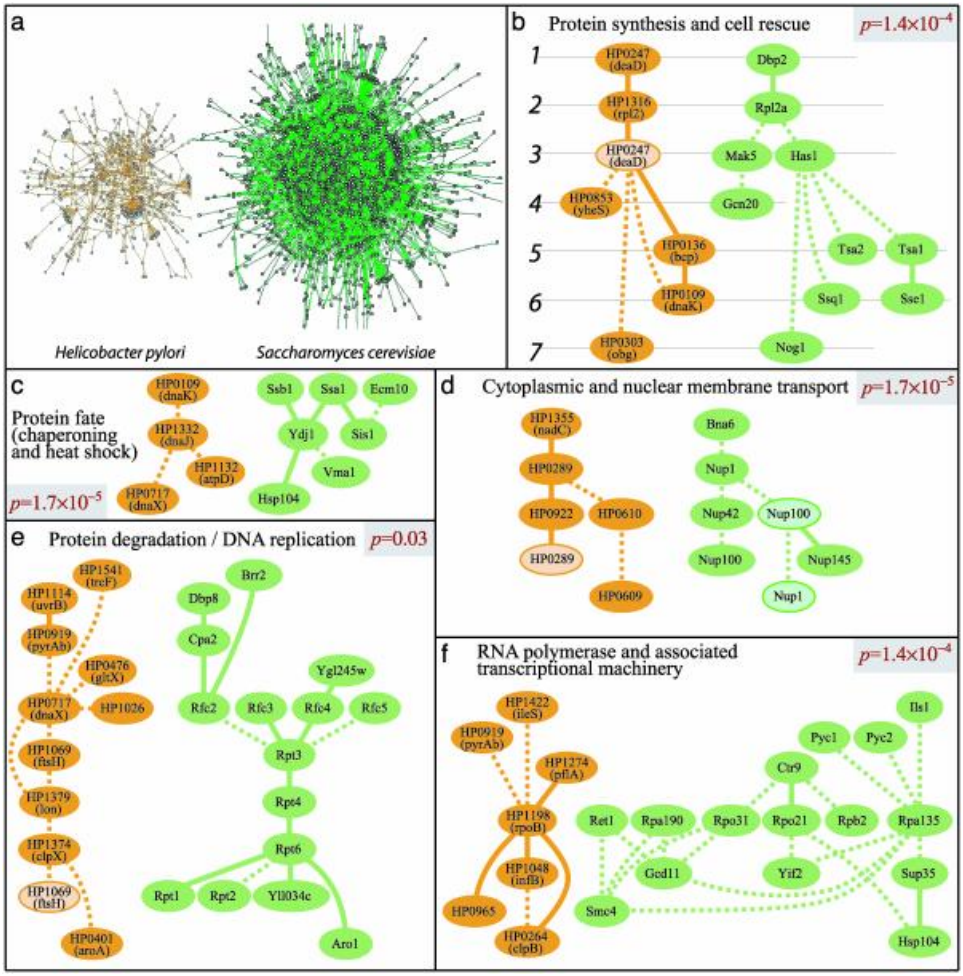


Figure 2.4: Top-scoring pathway alignments between bacteria and yeast. (a) The protein-protein interaction networks of *H. pylori* and *S. cerevisiae* were globally aligned to reveal conserved network regions through b-f processes. This figure is from [5].

work alignment is generalized to the multiple network alignment in Graemlin [6]. Figure 2.5 describes the Graemlin algorithm with four network examples, extracted from [6]. Graemlin algorithm merges two networks into one aligned network for the next step. Therefore, the order of alignment is important, which was selected by phylogenetic closeness. When align two networks, each network produces a number of d-clusters. The clusters for two networks are paired and scored using parsimonious ancestral history and BLAST score for proteins. The highest pair of d-clusters are selected as a seed and it is extended using a greedy algorithm. The result of alignment of two networks is three networks: one is aligned and merged network and the other two are unaligned

networks. The three networks are moved to the next step to be compared with a new network. Figure 2.5, extracted from [6], describes the procedure of Graemlin.

Network Querying Network Querying is another network comparison and it differs from network alignment as it searches a part of network or sub-network from a given network. PathMatch and GraphMatch are introduced in [7] for network querying. PathMatch finds a match with a given linear path. GraphMatch search a matching subgraph using a score function. The algorithms, however, have limitations as the scoring functions have unrealistic assumptions about mismatch and indels. Figure 2.6 shows an example by PathMath algorithm [7].

2.1.2.2 Algorithms

Traditional process of network motif discovery is divided into three steps: 1) Find all non-isomorphic subgraph patterns and record the frequency of each pattern in the target network. 2) Generate a number of random graphs, typically more than 10,000 graphs, and count the frequencies of each type. 3) Determine network motifs based on its statistical significance computed with the result of 1) and 2). The first and second tasks involve graph isomorphic testing which sits between P and NP-problem. Practical approach to the graph isomorphic problem is to label each subgraph with canonical labeling technique. Several algorithms exploited McKay's [2, 95] *Nauty*-algorithm for the labeling. Therefore, the main challenge of motif detection is to search all of the k-node subgraphs in the original and generated graphs as the search increases exponentially with the size of network as well as the size of subgraph. Previous algorithms present different approaches with different categories. For example, the algorithms are divided into exact counting and sampling algorithms, or, network-centric and motif-centric algorithms. To overcome computational infeasibility and incorrect results, parallel or distributed search algorithms are also developed.

Exact counting algorithm Network motif was first introduced as a functional building block in a transcriptional regulatory network by Milo et al. [16]. The authors fix the size of network motif as 3 or 4, and enumerate all the subgraphs of the size. This algorithm is called as an ERS

(Exhaustive Recursive Search) algorithm. It uses an $n \times n$ adjacent matrix as a data for n-node network, then enumerate all $\binom{n}{k}$ number of k-size subgraphs. Because of the expensive computing time, the motif size should be limited as less than 5.

For more effective search, Wernicke [15] developed an ESU (Enumerate Subgraphs) by building a search tree where the leaves are all subgraphs of the given size. Each child node of the tree is extended by adding one neighbor vertex to the set of vertices from the parent's node. During the search, it keeps an auxiliary dynamic list of nodes that are candidates for future additions. The authors also presented a sampling algorithm, RAND-ESU, which skips random set of branches in a tree during the search. Consequently, RAND-ESU improves the speed of search greatly and increases the feasible motif size up to 14.

On the other hand, Parida [17] introduced a compact graph representation for the search of k-node subgraphs. A compact graph representation is obtained by collecting an indistinguishable set of nodes together, and a motif is defined as a subgraph which appears frequently in a given threshold. The algorithm reduces the search time in a great amount as the number of representation enumerated from this compact graph is much less than the number of all subgraphs.

All the aforementioned algorithms are network-centric methods as they search graphs in the target network. On the contrary, Grochow and Kellis [96] presented a motif-centric method, which first lists all the possible patterns of given size as a query set, then search isomorphic subgraphs to each query in a network. This method was able to find up to 15-size motifs. However, the motif-centric algorithm can be inefficient for larger size of motifs, since the number of all possible query subgraph patterns increases exponentially with the size of motifs, which eventually results in many redundant searches. Table 2.1 shows the number of non-isomorphic directed and undirected graphs with up to 10 vertices, and we can see that the number of non-isomorphic graphs increases exponentially as the size of vertices increases. For example, the motif-centric algorithm needs to list 11,716,571 number of subgraph patterns in order to search for size 10 network motifs, and most of them might not even exist in the target graph at all.

Table 2.1: Number of Non-isomorphic Subgraphs for undirected and directed graphs with up to 10 vertices [1]

	Number of Non-isomorphic Subgraphs	
Vertices	Undirected	Directed
1	1	1
2	1	2
3	2	13
4	6	199
5	21	9,364
6	112	1,530,843
7	853	880,471,142
8	11,117	1,792,473,955,306
9	261,080	13,026,161,682,466,200
10	11,716,571	341,247,400,399,400,000,000

Approximation algorithms Exact counting algorithms can be infeasible, especially for larger size of motifs or in a larger size of target graph. Figure 2.7 shows that search time increases exponentially when the motif size increases, and Figure 2.8 indicates that search time also rapidly increases when the target network becomes larger. Therefore, many approximation algorithms have been introduced to reduce the search time and still produce as good results as with exact counting algorithms.

Kashtan et al. [33] presented a sampling algorithm and developed a tool, MFINDER. A fixed size of subgraph is sampled by extending adjacent edges starting from a random edge. Each sampled subgraph is weighted according to its edge selection probability and each type of subgraph is scored with the sum of weights of corresponding samples. MFINDER is scalable with the overall size of the network, but this method scales up to only 8-node network motifs. In addition, sampling bias and computing the weights of all samples are problems with this tool.

RAND-ESU [15, 18] is a randomized ESU algorithm. Like ESU, RAND-ESU builds a tree to search subgraphs of given size, but it visits only part of branches of the tree. Each level in the tree is assigned a probability to be selected, and the weight of each sample is computed with the product of probabilities. FANMOD [18] is a tool implementing ESU and RAND-ESU algorithm.

RAND-ESU reduces the sampling bias which was addressed with MFINDER, but it requires predetermined probabilities which produce inconsistent results.

NeMoFinder [19] is another approximation algorithm, which searches motif of size 3 up to given k. It first searches for repeated (frequency is larger than a threshold) trees of size $k-1$ and use the repeated tree to partition the graph. For each tree in each set of the partition, size -k subgraphs with $k-1$ edges are generated then extended to size-k subgraphs with k edges, filtered with threshold frequency again, and extended to $k+1$ edges, and extended to size $k+1$-subgraphs, etc. NeMoFinder algorithm is further upgraded into LaMoFinder [29] where network motifs were clustered further with a border informative functional class (FC) of GO terms as similarity measurements. Similar to LaMoFinder, there are some literatures [97] that attempted to assign biological significance to network motifs such as, motif mode [28] or motif theme [98]. A motif mode is defined as a combination of network motif and GO terms, and a motif theme is composed of some network motif instances with a biological meaning.

Kuramochi and Karypis [99] considered to find edge-disjoint subgraphs, using a priori-based motif detection algorithm which assumes that if a graph is frequent enough, then its subgraphs are also frequent. Therefore, maximum independent sets are first determined to find frequent edge-disjoint subgraphs. The work is only indirectly connected to network motif finding as the frequent subgraphs are defined with its high frequency rather than the statistical significance. The study is also interesting as it introduces the capability of finding disjoint network motifs.

On the other hand, Berg et al. [100] gave a different view for network motif, where a 'probabilistic network motif' is defined as 'similar' subgraph patterns. The authors asserted that as a network evolution is stochastic process, functionally related motifs are not necessarily isomorphic. Therefore, they built a statistical model for the occurrence of such motifs and derived a scoring function measuring the statistical significance. Search for topological motifs was conducted based on this scoring function and a graph alignment.

Parallel search for network motifs Exact counting is infeasible and approximation search can lead a false detection. Therefore, parallel search for exact counting might be an in-

evitable solution. As an example, Wang et al. [20] partitioned the target network without overlapping for an easy parallel search. The algorithm was experimented with an *E. Coli* transcriptional regulatory network and found up to 5-node network motifs.

Schatz et al. [21] also conducted a parallel network motif finding. The project mainly used motif-centric method proposed by Joshua and Kellis [96]. First, the target network was divided into smaller networks allowing overlaps. All query subgraphs of given size were listed, then isomorphic graphs of each query in each smaller network was counted in parallel. Because the sub-networks are overlapped, repeated subgraphs can happen. In that case, the vertices are stored in a central hash set to avoid double counting. Two parallelization methods were described: *query parallelization* and *network parallelization*. Query parallelization assigns a subset of query subgraphs to each worker, and each worker has an access to the whole target network to search the assigned subgraphs. On the other hand, network parallelization is good for searching a single query subgraph in a network. In this parallelization, a target network is divided with overlaps, and each sub-network is assigned to each process. Each worker searches the whole query subgraphs in the assigned sub-network. Obviously, a main issue in network parallelization is how to divide the target network so that it can avoid repeated counting for the same subgraph.

2.1.2.3 Applications

Network motifs have been used for various biological applications. The concept of network motif is originated from a gene-regulatory networks of *E.coli* [88], and the properties of motifs are analyzed. Feed-forward loop (FFL) and Bifan motifs are the most common motifs discussed in many researches [101, 101, 102]. For example, gene expression noise of all possible circuit architectures of feed-forward loop (FFL) motifs are investigated and the results show that FFL architectures have two functional categories based on its ON/OFF states in [103]. Bhardwaj and Lu [104] discovered that a more connected hub or network motif in the interaction network is expected to be more important. Since the complexity of a motif reflects its essentiality, the results show that more important motifs have an increasing congeniality between their components as compared to random motifs.

While network motifs are commonly analyzed for static properties, Prill et al. [105] analyzed the dynamic properties of network motifs which contribute to biological network organization. In the study, network motifs are categorized into three classes: stable motifs, moderately stable motifs and unstable motifs based on the structural stability score, and this concludes that robust dynamical stability of network motifs determines the network's dynamic property.

Hallinan and Wipat [106] investigated the dynamic property of networks via network motifs as well. They investigated the correlation of network motifs and oscillatory dynamics in a yeast transcriptional network and a set of computational networks. Experimental results showed that network motifs are not vital to network dynamics in context because of the tight interconnection with other components. This observation suggests that the classic motifs are not necessarily involved in the generation of oscillatory dynamics in biological transcriptional networks.

Network motifs are also used for predictions. The presence of conserved interaction of motifs within the network helps to predict protein-protein interaction in [26]. Protein function is also predicted using network motifs in [29]. In this study, network motifs are further categorized with enriched GO term, and the GO term combined network motifs helps predict a protein function better.

Another application of network motifs was identifying some specific-type of genes. 3-node network motifs were applied to recognize breast cancer susceptibility genes in [107]. Each network motif is ranked based on an activity score, and a number of motifs are selected as markers given a threshold. The markers are the input features for a support vector machine to predict breast-cancer related genes.

On the other hand, network motifs were used for building a higher structure of a graph. First, a 'motif cluster' is introduced in [108], then the instances of Feed-Forward Loop (FFL) and Bifan motifs are aggregated into homologous motif clusters and many of the motif clusters are overlapped with known functional modules. In addition, the combination of two motif clusters composes a core network, playing a critical role in the global structural organization. Or, motifs were generalized in the paper [109] by combining various size of network motifs. An efficient algorithm

for the detection of network motif generalization was provided and experimental results showed that generalized motifs can be different even with common motifs.

Zhang et al. [98] tried to validate network motifs in an enriched form. Five types of different interaction networks were combined , then 3-node and 4-node network motifs were divided into a number of 'motif themes' based on the elements. Through the work, the authors asserted that motif themes are more appropriate to represent fundamental network design principles. The work is a different version of motif generalization and it is also similar to the work of LaMoFinder as the network motifs are clustered further with different biological meanings. The biological relevance of a motif theme is often much clearer than the relevance of the underlying motifs. However, this paper lacks a comprehensive analysis for all instances of motifs because only some examples are provided and analyzed.

Network motifs were used to distinguish certain networks as a distance measure in [23], or as classifiers [24]. The frequency of network motifs are defined as the 'relative graphlet frequency distance' in [23], and it was applied to PPI networks of yeast and fruit fly. As a result, geometric random networks can model PPI networks better than scale-free networks. The motif frequencies are used as classifiers in [24], which selected an artificial model similar to the PPI networks, by plugged into machine learning techniques. In [110], the motif frequencies is defined as 'MotifScore' to score the molecular docking, where the motif size is limited to five.

Another application was using motif frequency profiles to cluster the network into different super-families [25]. Size 3 to 4-node motifs are used for directed and undirected networks respectively. It is also interesting to see that, for undirected networks, the statistical significance of network motifs is not considered, since they are highly dependent on the size of network. The statistical measurements, therefore, were questioned in [111]. The size of network motif is also questioned as there is a chance to improve the discrimination performance by increasing the motif size further [105].

Evolutionary conservation of motifs can be investigated as well. In gene-regulatory networks of *E.coli* and *S.cerevisiae*, the genes of different motifs are not evolutionary related [27], indicating that the instances of the motifs are not copied from ancestral circuits. However, network motifs

in a PPI network show significantly higher portion of relations to the known orthologies of five eukaryotes [97]. Therefore, they concluded that motifs may represent evolutionary conserved topological units of cellular networks which is constructed with the specific biological function where the motifs participate. In addition, groups of proteins in a highly interlinked cluster tend to be conserved in a cohesive group if they represent an evolutionary conserved functional module. Each motif is assigned to a functional class to examine the relationship of function and evolutionary conservation in motifs. Experimental results showed that larger motifs have a notable functional homogeneity. Consequently, the conservation rate of motif members was much higher, suggesting that we should focus on a group of proteins, not just a single component.

Lee et al. [28] investigated network motifs in PPI beyond the score of motif topology, and introduced a concept of 'motif mode'. If the subgraphs of the same topology has different evolutionary constraints, then they are categorized into different motif modes. The authors uncovered up to one million motif modes each of which features a unique topological combination of molecular functions in GO term. The process is similar to LaMoFinder [29], as it first determines network motifs using statistical measurements, then divide them further by GO molecular function terms. While LaMoFinder developed a distance measure using GO terms in order to cluster the motifs, 'motif mode' is clustered according to corresponding GO molecular functional terms of depth 5 and depth six, resulting millions of motif modes. The search of motif mode can be faster with an intelligent agent-based distributed computing [112].

Furthermore, network motifs of size 3 or more are used as an elementary unit of organization [113]. Network motifs were used to assess the relationship between the transcription regulatory network and chromosomal organization in *E. Coli* and in a *budding yeast*, yielding significant biological insight. In the studies [88, 108, 114], motifs sometimes appear in clusters which is not separable.

2.1.2.4 Issues

There are several problems regarding network motifs through a number of algorithms and applications. We discuss the problems in this section and deal with the problems at the later chapters.

How to qualify network motifs beyond the structural uniqueness? Traditional definition of network motif is a over-represented k-subgraph pattern with statistical significance against random networks. Although network motifs have been studied in biological networks, the work focusing on how to measure network motifs in biological sense is rarely done. In fact, Hallinan and Wipat [106] asserted that network motifs themselves do not provide contextual information in biological networks. Ingram et al. [102] claimed that since a structure itself is insufficient to understand function of gene-regulatory networks, additional information is necessary. These suggest that we need other validation criteria to measure biological quality, as statistical importance alone is incapable of catching the properties of network motifs.

Are all of the instances of network motif are significant? Network motif is defined as an over-represented subgraph pattern, meaning each pattern has a number of instances. After all those expensive search in the original as well as in random graphs, existing tools [16, 18] report only the numbers of each pattern. Therefore, the tools are not directly useful in the applications where the elements of motifs are more important. For example, the applications [105, 107] first decide the pattern of network motifs, then should search again for the instances of each network motif to apply. But if we will use only a subset of network motif instances, then how much of them are useful and how to choose the 'useful instances'? In extreme cases, most of them are useless in some applications. Therefore, additional pruning methods in the early steps would be more sensible.

Structurally insignificant types are not biologically insignificant? Habibi et al. [115] used k-connected subgraphs, which are not network motifs, to predict protein complexes.

This study shows that it is possible that structural uniqueness is unrelated to biological functions. In fact, we could not find any evidence that non-motifs are useless in applications [28, 29, 97]. Instead, non-motif instances are simply not considered in those applications. Therefore, in this thesis, we examine those non-motifs in order to connect them with biological properties, just to open some possibilities of the non-motifs' usages to applications.

What roles do network motifs have? Sequence motifs have examples of biologically validated roles, such as, DNA transcription factor binding sites or zinc finger protein motifs. Biologically validated sequence motifs have been categorized and stored in many databases, based on their structural or functional roles. However, biological roles for network motifs are still unclear, and there are no databases. There have been some efforts to discover the roles of network motifs: They have been used to construct larger structures, such as motif cluster or motif generalization [108, 109], to get more meaningful results; Wang and Provan [116] showed that network motifs exist between functional modules which is not separable, contradicting the early claims that network motifs are building blocks of functional modules. Therefore, we believe that what biological roles network motifs have in biological systems is still an open question and studies should be further focused on this matter than the computational problems.

Besides, there are still other issues to be seriously considered. First question is related to the size of network motifs. Will there be any *optimal size* of network motifs? How can we decide the size of network motif to apply in a real problem? In [19, 96], larger size of network motifs are effectively searched using symmetry-breaking technique or extending the motif size from basic tree structure. However, it is still a question if larger network motifs are more meaningful than smaller ones. Second issue is its background knowledge. We have used gene ontology information to see any biological properties of network motifs. There must be other information that are closely related to network motifs. We believe those questions will motivate interesting future studies for network motifs.

2.2 Biological Network

In bioinformatics, network motifs are studied in biological networks which are developed with the advent of systems biology. Therefore, we review biological networks in the context of systems biology in this section. In fact, systems biology is the study of integration which has two aspects [117]. One is the study of patterns within a cell or an organism while the other is the comparison across difference species. We take the first aspect of integration study of systems biology in this thesis.

Systems biology is defined as a system-level understanding of biological systems [118], and components of which include molecules, cells, organisms or entire species. It is the study of interaction of biological components, and aims to appreciate entire biological systems with illuminating, modeling, and predicting the behavior of biological elements and their interactions [13]. Denis Noble [119] described systems biology as a study of integrating individual components rather than reduction, and Chong and Ray [120] defined it as a whole-istic approach. Ever since systems biology has been developed with the collection terms of *-ome* such as genome, proteome in 1990s, more than hundred of *-omics* technologies have been defined as described in [121]. We should emphasize that rapid growing of biological knowledge, huge amount of data in database and advances in computational tools excelled the development of systems biology. However, the biological data so far still requires experimental devices, advanced software and analytical methods [122]. As an effort to analyze system level biological data and understand the integration of molecules in the biological context, the -ome data are represented as biological networks and the biological systems are modeled and simulated via the networks.

2.2.1 Types of Biological Networks

Biological network is an efficient way of extracting useful information out of the huge biological data [13]. A biological network consists of a number of vertices representing molecules (DNA, RNA, proteins and metabolites) and edges for their interactions. Built only with its local relationships between two nodes, biological networks provide a global view to understand the

whole structure. There are various types of biological networks based on different molecules consisting of the nodes, and we will review gene regulation network, metabolic and protein-protein interaction networks as the followings.

2.2.1.1 Gene Regulation Network

A gene regulatory network is a complex combination of different types of sub-networks including signaling transduction pathway, transcriptional regulatory network and intercellular molecular regulatory networks, each of which performs a physical, chemical and functional processes [13, 123]. Regulation networks are critical factors for evolutionary changes and organism life. The vertices are a number of DNA segments and edges are generated by various interactions between them. Figure 2.9, which is from [8], shows an example of a gene regulation network in a mammalian cell. A number of databases such as KEGG [124], EcoCyc [125], GeneNet [126], RegulonDB [127] and TRANSFAC [128] provide gene regulation networks.

Signaling Transduction Pathway A signal transduction pathway is the process where an extracellular signaling molecule binds to a membrane receptor then the receptor conveys a signal inside the cell [129]. Sometimes the signals occur within the cell, amplifying to a large response [130]. The signal creates gene expression or enzymes activation in the cytoplasm. The triggered transcription factors by a signal in turn bind to the regulatory regions of genes to activate gene expression [13]. That is, the targets of signaling transduction pathway are metabolic enzymes and transcription factors, which in turn generate transcription regulatory networks. Figure 2.10 shows signaling pathways in yeast, from the paper [9].

Signaling pathways are relatively small as it uses a more confident experimental results than other networks. Note that not all protein-protein interactions react by chemicals and not all components in signaling pathway are proteins. In topological structure, a signaling pathway follows power-law and small-world properties. Signaling pathway data can be found in the literatures [131, 132] or databases such as KEGG [124], EcoCyc [125], TRANSPATH [133, 134] and GeneNet [126].

Transcription Regulatory Network A transcription regulatory network is the set of transcription factors and the genes that they bind [87]. There are multiple relationships between transcription factors and the genes: multiple transcription factors can regulate one gene; one transcription factor can regulate multiple genes. Signaling transduction often takes transcriptional networks as targets [127, 128, 135]. The transcription regulatory network of *S. cerevisiae* [136] is one of the first data generated and network motifs were first introduced in *E. Coli* transcriptional regulatory network [88]. Network motifs are also aggregated into a larger structure in this type of network [108] and parallel algorithm for network motif discovery is conducted with this network as well. An example of a transcription regulatory network in a cell is depicted in Figure 2.11 which is from the paper [10]. Transcriptional networks also follow power-law and small-world topological properties. Similar to the signaling pathways, the main source of transcriptional regulatory networks are experimental literatures. Databases of transcription regulatory networks include KEGG [124], EcoCyc [125], TRANSFAC [128,135] and RegulonDB [127]. The networks are built on a single organism, such as *S. cerevisiae* [136] and *E. Coli* [88].

2.2.1.2 Metabolic Networks

Metabolic networks are composed of metabolites (glucose, amino acids, polysaccharides and glycan) and their biochemical reactions [13]. A metabolic pathway is a small local area of a metabolic network and a metabolic network provides more comprehensive view of the cellular metabolism [11, 13]. The network determines physiological and biochemical properties of a cell and one example of metabolic network is provided in Figure 2.12 which is from [11]. Some examples of metabolism include *Aerobic Respiration*, *Anaerobic Respiration*, *Carbohydrate metabolism* and *Lipid metabolism*. While most of metabolic networks are manually inferred from literatures, Francke et al. [137] introduced an algorithm for metabolic network reconstruction. The databases providing metabolic networks are KEGG [124], EcoCyc [125], GeneNet [126], MetaCyc [138] and BioCyc [139].

2.2.1.3 Protein-Protein Interaction Networks

Proteins are vital parts of organisms, provide structural or mechanical functions and participate in every process within cells such as cell signaling, immune responses and the cell cycle. Proteins are complete biological molecules in a stable conformation and are made of twenty possible amino acids arranged in a linear chain. The chemical interactions of amino acid residues determine the conformation of proteins and form a relationship between protein sequence and structure. A protein sequence defines three dimensional structure of the protein and the structure determines its function.

Proteins are functionally more active by interacting with other molecules of DNA or other proteins. Protein-protein interactions (PPI) carry out a number of cellular processes such as multi-enzyme complex, signal transduction chain, protein scaffolds and basis for the function of ribosomes [13]. Therefore, molecular level of studies of proteins have limitations for clear insight of biological function, which led to a system level of study. Protein-protein interactions are in fact the core system in a living cell and rapidly growing data sets. Many advanced technologies helped establish complete protein-linkage maps [140].

Currently existing protein-protein interaction data sets are generated with various methods including experimental techniques and computational methods. Phizicky et. al [141] presented a number of experimental methods to detect and analyze protein-protein interactions, which are physical, library-based and genetic approaches.

Physical methods are to select and detect proteins binding another protein using protein affinity chromatography, affinity blotting, immunoprecipitation and cross-linking. Affinity chromatography is a method to separate biochemical mixtures and detect a highly specific biological interactions. Affinity blotting uses an antibody as the probe and immunoprecipitation precipitate and wash a protein antigen using an antibody to purify protein. Cross-linking infers protein-protein interactions by deducing the architecture of proteins or probing extracts, whole cells or partially purified preparations.

Library-based method is to screen large libraries for genes or gene fragments for possible protein interactions. Protein probing, phage display and two-hybrid systems belong to this method. In protein probing, a labeled protein is selected for a probe to screen an expression library to identify proteins interacting with it. Phage display uses bacteriophages to connected proteins with the genes encoding them. Two-hybrid system [142] detects protein interactions using molecular organization of many transcription factors.

Extragenic suppressors, synthetic lethal effects, overproduction phenotypes and unlinked non-complementation belong to the genetic methods. With extragenic suppressor analysis, new mutations can be discovered and an analysis of the genes and proteins defined by these mutations sometimes indicates interacting proteins. Synthetic lethal effects use a synthetic effect where mutations in two genes can cause death while mutation in either alone does not. This phenomenon can detect physical interactions between two proteins required for the same essential function. Since the overproduction of some proteins can provide insight into protein-protein interactions, the overproduction phenotypes are also used for the detection. Unlinked non-complementation uses the case where individuals, heterozygous for two different recessive mutations, sometimes display a mutant phenotype.

In addition to experimental methods described above, many computational approaches are used to predict protein interactions [143]. The computational methods use presence or absence of genes in related species, conservation of gene neighborhood, gene fusion events, similarity of phylogenetic trees and *in silico* two-hybrid method to predict protein interactions. In silico two-hybrid method uses the relationship between correlated residues and interacting surfaces based on the differential accumulation of correlated mutations between the two proteins. The protein-protein interaction (PPI) network in Figure 2.13 is built by combining the local pair of interactions into global maps.

Because of the various detection methods, PPI networks are various and noisy as well. Generally, PPI has scale-free and small-world properties. The property of scale-free nature explains some interesting aspects of PPI network. High percentage of hubs of PPI play important roles as essential proteins or evolutionary conserved proteins [144]. However, the scale-free property

is arguable as Przulj et al. [23] showed that PPI networks better fit to random graph model than scale-free model.

A number of databases provide PPI data sets and Fischer et al. [145] described the databases for protein interactions. DIP [146], BIND [147], MINT [14], KDBI [148] and BindingDB [149] databases are comprehensive interaction databases providing protein-protein, protein-nucleic acid and protein-ligand interactions. The sources are from experimental methods and data-mining of literatures. Fischer et al. [145] categorized the databases for protein-protein interactions only as the followings: Organism-specific databases, structural/mutational databases and general databases. The Comprehensive Yeast Genomic Database (CYGD) [150] and the Human Protein Reference Database (HPRD) [151] are the organism-specific databases. Structural or mutational databases contain ASE database [152] and BID [153]. Other databases such as GRID [154] does not follow any specific category.

2.2.2 Network Property

Biological networks show some topological properties compared to other random networks, which are scale-free and small-world [155]. A scale-free network is one whose degree distribution is characterized by a power law of the form $P(k) \sim k^{-\gamma}$ [156]. A small-world network is one where any two vertices in the network can be connected by a relatively short path, 6 distance. [157]. But, these properties are not just for biological networks. A number of real-world networks, such as the Internet and citation networks have similar properties [158]. And there are some disputes of biological networks being scale-free network [159]: 1) Biological networks are incomplete or noisy and; 2) Depending on the methods to build the biological network, scale-free network is not applicable. For example, PPI network is more like a random network than a scale-free network; 3) Different graph representation can also have different conclusion for small-world and scale-free for the metabolic network in E. coli. [160]. 4) Arita [161] challenged the scale-free property by claiming that it does not give any meanings for biological insight. He and Zhang [162] explained that hubs are essential, simply because they have more interactions than others.

2.2.3 Challenges

Systems biology is the study of whole and integration. Biological networks are the representative for the study of systems biology. Although each biological network is constructed separately, the networks have many overlaps, so changes of one network can affect other networks as well. Therefore, one of challenges in systems biology is to integrate this data for thorough analysis [87]. Next challenge is to incorporate other biological information into biological networks. Three-dimensional structure of proteins [163], temporal and spatial location [163–166], functional or evolutionary context [167–169] and environmental conditions [170] have been included into biological networks for enriched studies. Although it is obvious that incorporation causes another computational challenges [171, 172], this is an appropriate and necessary step for better understanding of systems biology.

2.3 Granular Computing

Real-world networks, especially biological networks are huge and they are continuously growing. Trends for more data integration into networks further push more efficient computations in systems biology, and granular computing is one of many solutions. Granular computing [173, 174] involves the processing of complex information granules arising in the process of data abstraction and the derivation of knowledge from the data. In the case that little knowledge for data attributions is given, clustering the data sets is a proper method to realize granular computation. Therefore, we review various clustering algorithms in this section.

2.3.1 Data Clustering Algorithms

Jain et al. [12] defined the cluster analysis as the “organization of a collection of patterns into clusters based on similarity”. The difficulty lies in the definition and the scope of ‘cluster’ in the data set. In [175], a cluster is defined as 1) a set of similar objects (minimum-intra), or, 2) a set of points such that the distance between two points in a cluster is less than the distance between

the points of other clusters (between-cluster and intra-cluster), or 3) densely connected regions in a multi-dimensional space separated by loosely connected points (graph or dense-based).

Clustering analysis is distinguished with other analysis in the following criteria as described in [12, 176]. First, clustering analysis is an unsupervised classification. Unlike supervised classification whose goal is to assign an input data into one of the classes based on the classification learned with labeled training data [177], unsupervised classification is to separate or partition unlabeled data into several clusters based on the *conceptual* or *hidden* properties of the input data sets. Second, clustering analysis is an unsupervised 'non-predictive' learning method which divides the data sets into several subsets on their subjective measurements, while an unsupervised 'predictive' learning is based on the 'trained characterization'.

Various data clustering algorithms are categorized into various ways as we can see in the papers [12, 113, 176]. Jain et al. [12] provided the taxonomy of the clustering algorithms in a hierarchical structure as depicted in Figure 2.14. Because of the different aspects of the clustering algorithms, categorizations of existing clustering algorithms are also various as we can see in the survey papers of clustering algorithms [12, 113, 176]. Andreopoulos et al. [113] divided the whole clustering algorithms into 6 categories: Partitioning, Hierarchical, grid-based, density-based, model-based and graph-based clustering algorithms. Here, we omit the discussion of grid-based algorithm. Instead, we add another category, evolutionary clustering algorithms, described in [12]. Note that this categorization is a soft classification as some of them can belong to several groups.

2.3.1.1 Partitioning Clustering Algorithms

Partitioning clustering methods are useful for the applications where a fixed number of clusters are required and Andreopoulos et al. [113] further divided it into numerical methods and discrete methods. K-means algorithm and Farthest First Traversal k-center (FFT) algorithm [178], K-medoids [179] or PAM (Partitioning Around Medoids), CLARA (Clustering Large Applications) [180], CLARANS (Clustering Large Applications Based Upon Randomized Search) [181] and Fuzzy K-means [182] belong to numerical methods. On the other hand, discrete methods in-

clude K-modes [183], Fuzzy K-modes [184] etc. K-means algorithm and K-mode algorithms are the most common methods. K-means algorithm [185] iteratively assigns the objects into one of cluster whose center is the closest, with k number of initial mean vectors. The process continues until there is no other re-assignment or it depends on the user-specified threshold. Huang [183] provided k-modes algorithm which extends the k-means methods to categorical domains, as k-means only deal with numerical data sets. Recently, a nonnegative matrix factorization (NMF) method is applied for clustering micro array data as we can see in [186]. With the sparse coefficient matrix factored from the original data set, the membership of each data can be assigned to one of the clusters.

2.3.1.2 Hierarchical Clustering Algorithms

Hierarchical clustering algorithms divide the data into a tree of nodes, where each node represents a cluster [113]. Hierarchical clustering algorithms are often divided into two categories based on their methods or the purposes: Agglomerative vs. Divisive; Single vs. Complete vs. Average linkage. In some applications including bioinformatics, hierarchical clustering methods are more popular as natures can have various levels of subsets. But hierarchical methods are slow, errors are not tolerable and information losses are common when moving the levels. Like partitioning methods, hierarchical methods consist of numerical methods and discrete methods. BIRCH [187], CURE [188] and Spectral clustering [189] are numerical methods while ROCK [190] and LIMBO [191] are discrete methods.

2.3.1.3 Evolutionary Clustering Algorithms

Evolutionary approaches use evolutionary operators (such as selection, recombination and mutation) and a population to obtain the optimal partition of the input data [12]. The first step of these algorithms is to choose a random population of solutions, which is usually a valid partition of data with a fitness value. As a next step, they use the evolutionary operators to generate the next population. A fitness function, which determines a population's likelihood of surviving into the next generation, is applied to the solutions. The two steps are repeated until it finds the required

solution meeting some conditions. Generic algorithms (GA) [192] and evolution strategies (ES) [193] belong to this category.

2.3.1.4 Density-based Clustering Algorithms

Density-based clustering algorithms use a local density standard. Clusters are dense subspaces separated by low density spaces. One of the examples is DBSCAN introduced in [194], which was developed to cluster large-scale data sets in the context of data mining. It requires that the density in a neighborhood for a data should be high enough if it belongs to a cluster. A new cluster from one data point is created by including all points in its neighborhood. The threshold of neighborhood of a data point is user-specific. DBSCAN uses $R*$-tree structure for more efficient queries. The authors showed the effectiveness and efficiency of DBSCAN using synthetic data and SEQUOIA 2000 benchmark date as well. Other density-based clustering algorithms include CLIQUE [195] and HIERDENC (Hierarchical Density-based Clustering) [196].

2.3.1.5 Model-based Clustering Algorithms

Model-based clustering uses a model which is often derived by a statistical distribution. AutoClass [197] is the most popular example of this category and it is based on Bayesian method for determining optimal classes in large data sets. In the probabilistic point of view, data points are assumed to be generated according to probability distributions. Combining it with clustering point of view, each cluster is represented with different probability distributions, (different type or different parameters). The algorithms belonging to this category mostly use expectation-maximization (EM) approach. It first initializes the parameters of each cluster, then computes the complete data log-likelihood in e-step and selects new parameters maximizing the likelihood function. AutoClass considers a number of families of probability distributions including Gaussian, Poisson and Bernoulli, for different data types. A Bayesian approach is used in AutoClass to find out the optimal partition of the given data based on prior probabilities.

2.3.1.6 Graph-based Clustering Algorithms

According to the paper [113], graph-based clustering algorithms were applied to interactomes for complex prediction and to sequence networks. Junker and Schreiber [198] reviewed some of graph-based clustering algorithms for bioinformatics applications. As biological data can be represented as a graph, network clustering algorithms use graph structure for the network. Authors also reviewed Clique-based and Center-based clustering techniques for small data sets. For the large data sets, it refers some techniques including distance k-neighborhood, k-cores and quasi-cliques as well. As these methods are closely related to the analysis of biological networks, we discuss them in more detail in the next section as network clustering algorithms.

2.3.2 Network Clustering

Network clustering or graph-based clustering algorithms deal with the data represented as a network or a graph. Data points are represented by vertices and an edge exists if two data points are similar or related in a certain way. Network clustering approaches are used to perform a distance-based clustering and conceptual clustering. In distance-based clustering, edges are generated by the *closeness* between data points. Conceptual clustering generates a concept of description for each generated cluster [13]. Network clustering has been applied to many researches: It has been used for understanding the structure and function of proteins based on protein interaction maps of organisms; Protein interaction networks are clustered using cliques to decompose the protein interaction network into functional modules and protein complexes [199].

Network clustering problem is to find subsets of a given graph such that each subset is a cluster modeled by structures such as cliques or other distance and diameter-based models. The clustering models are classified by the constraints on relations between clusters or the objective functions used to achieve the goal of clustering. Clusters may be allowed some overlaps or no overlaps, based on the applications. Network clustering algorithms come with two types of optimization problems: minimum number of clusters or maximum cohesiveness within each cluster. Clique-based clustering [200–202] and center-based clustering algorithms [203–206] have been

developed especially for network clustering. Clique-based clustering algorithms use the structural connectivity between vertices while center-based clustering algorithms use the similarity of the elements with the cluster's center. We should mention that most of network clustering algorithms are heuristic as exact algorithm is computationally infeasible.

We notice there are close relationships between network clusters and network motifs from the work in [202]. Most of network clustering algorithms partition the network without overlaps, and try to find the best matches with known structures, such as protein complexes or functional modules. However, recent researches [207–209] focused on finding overlapping clusters. If network motifs are over-represented and overlapping subgraphs, network motifs can be extended to network clusters in a big picture. Especially for the protein complexes of a very small diameter and a very small average vertex distances, small-sized network motifs are proper candidates for protein complexes or functional modules. In a sense, network motifs are more rigid than network clusters as the size of network motif is fixed while network clusters can have various size.

Furthermore, network clustering algorithms can be a filtering tool for biological network motif discovery algorithms. As some parallel algorithms first cluster a network for a faster search of network motifs, we utilize various network clustering algorithms for fast search of biological network motifs. In our methods, network clustering has two advantages: First, for an approximation search, we can reduce the amount of total search with clustering network motifs and reducing the potential subgraphs from the boundary edges between clusters. Secondly, the resulting clusters can be easily used for a parallel network motif search.

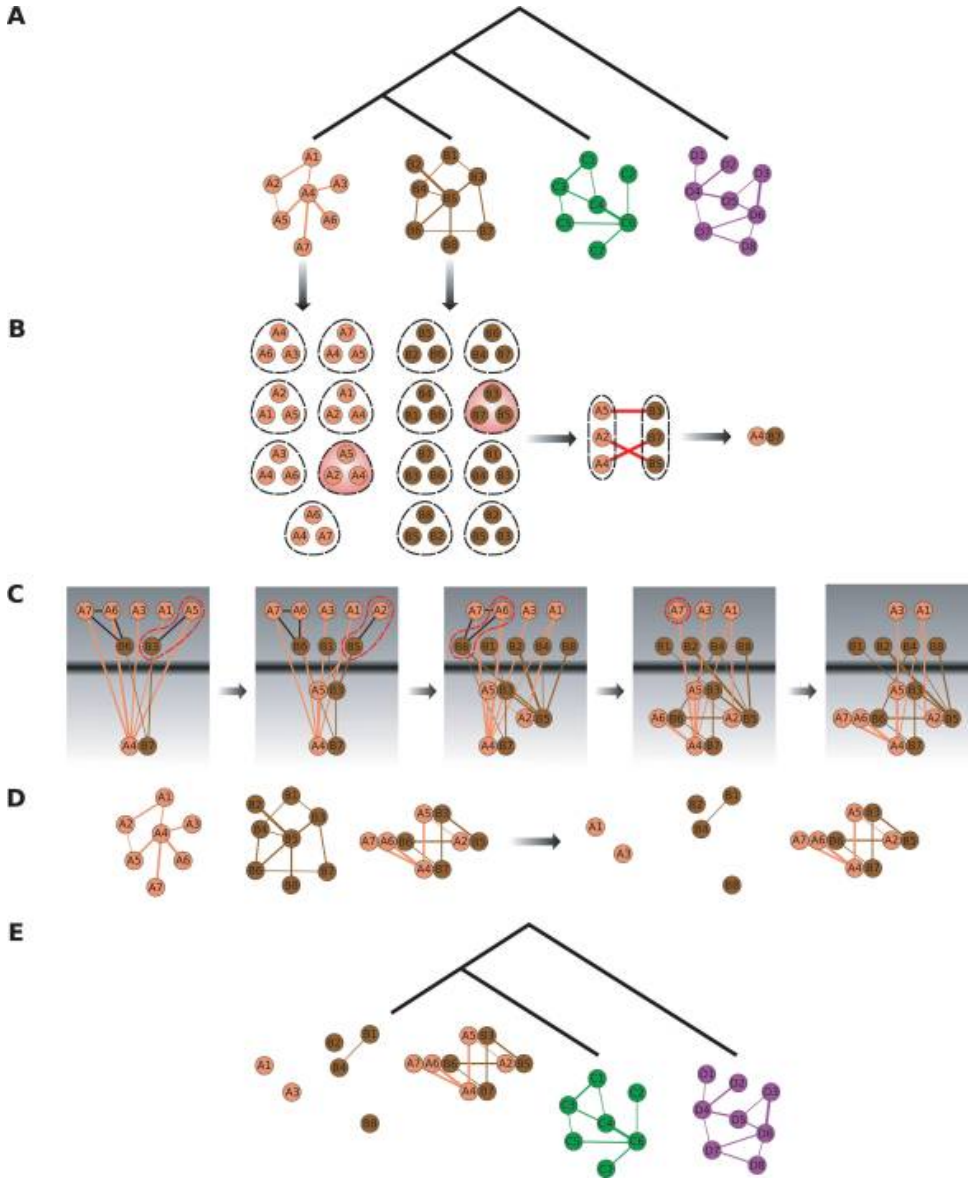


Figure 2.5: Outline of the Graemlin algorithm, by the courtesy of [6]. (A) Four networks with their phylogenetic relationships. (B) Graemlin first performs a pairwise alignment of the two closest species, using d-cluster and a pair of seeds. (C) Graemlin extends the seed using a greedy algorithm. (D) Graemlin transforms the resulting alignment and the unaligned nodes into three generalized networks for use in the next step. (E) In the next step, Graemlin will perform three pairwise alignments, one for each of the newly created generalized network.

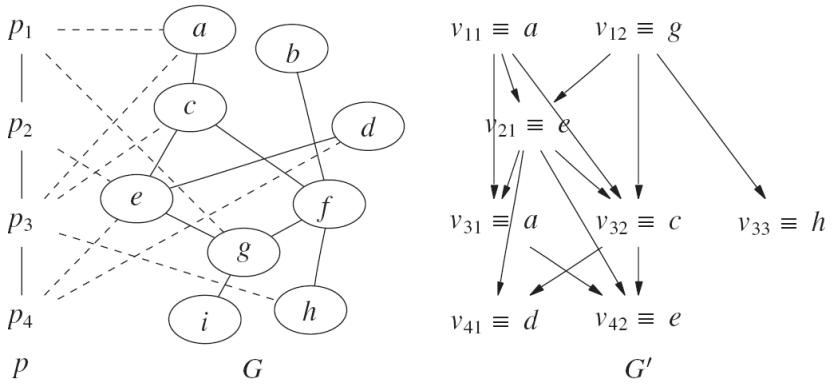


Figure 2.6: A query network g and a target network G are given in the left-side hand. The resulting graph G' is constructed in the right hand side, using PathMatch algorithm. Dashed lines show vertex correspondences and $\equiv$ in G' means the representing vertex in G. In this example, at most one mismatches or indels are allowed between two matches. Figure is from the paper [7].

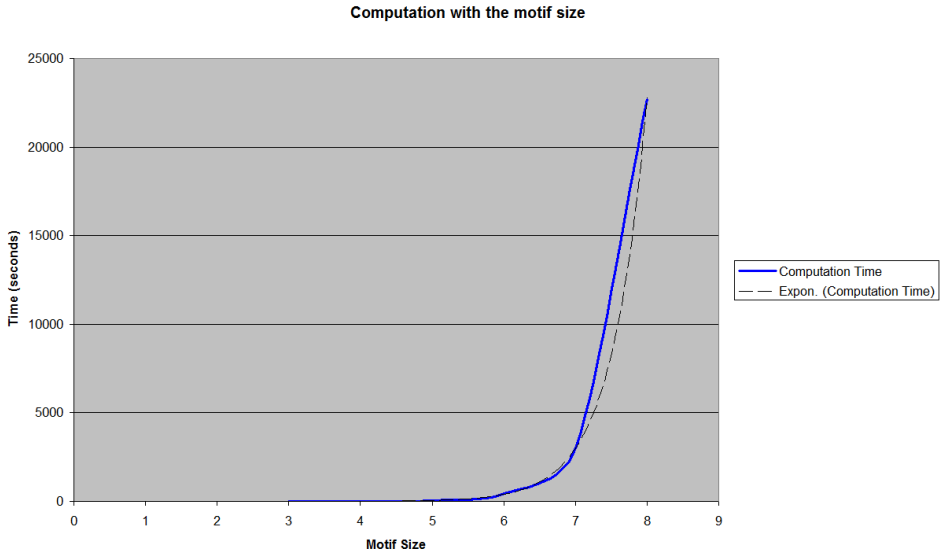


Figure 2.7: Subgraph search time increases rapidly as the motif size increases. The horizontal axis is the size of motifs and the vertical axis is the time consumed for the search. The dashed line is an exponential curve to show a trend of search time based on the size of motifs.

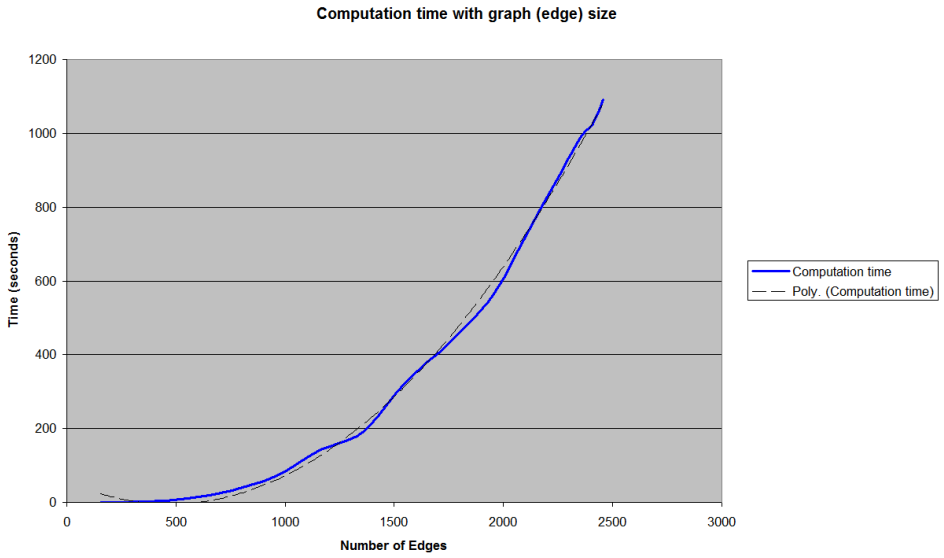


Figure 2.8: Subgraph search time increases rapidly as the size of a network increases. The horizontal axis is the number of edges and the vertical axis is the time consumed for the search. The dashed line is a polynomial curve to show a trend of search time based on the size of a network.

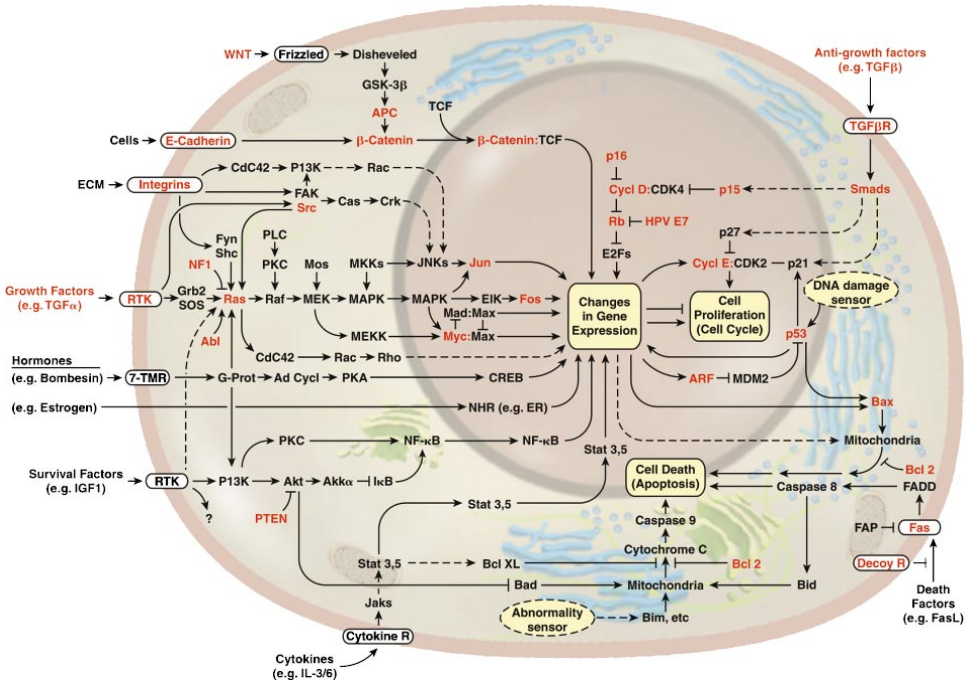


Figure 2.9: The emergent integrated gene regulation network representing the cell progress in a mammalian cell. The signaling pathway has begun to lay out a circuitry that will likely mimic electronic integrated circuits in complexity and finesse. Gene expression process has much overlap regions with signaling pathways. The figure is from [8].

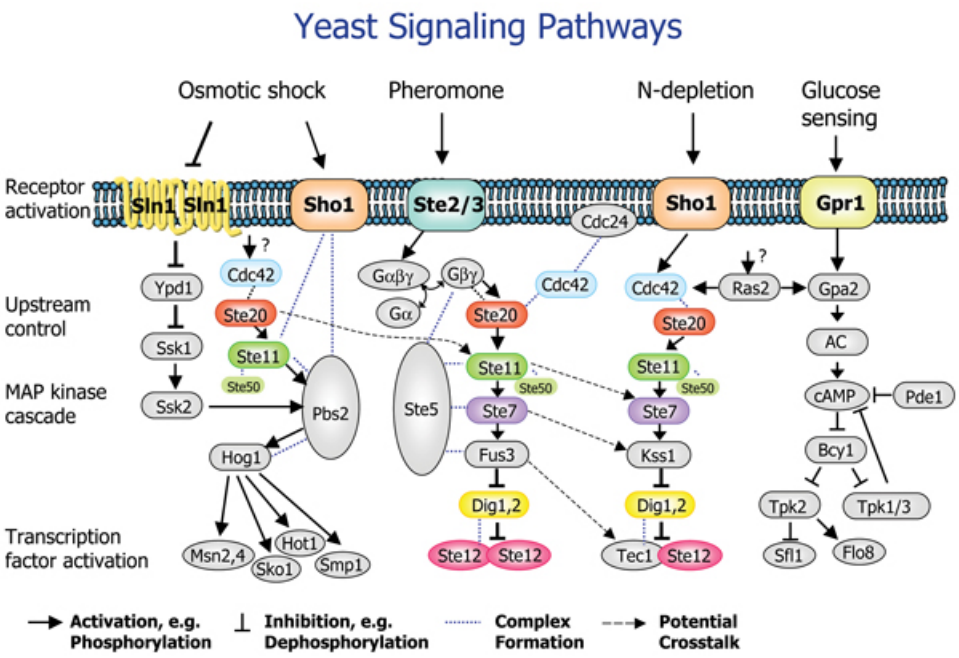


Figure 2.10: Overview of signaling pathways in the baker's yeast *S. cerevisiae*. The activated receptor activates intracellular processes. The figure is from [9].

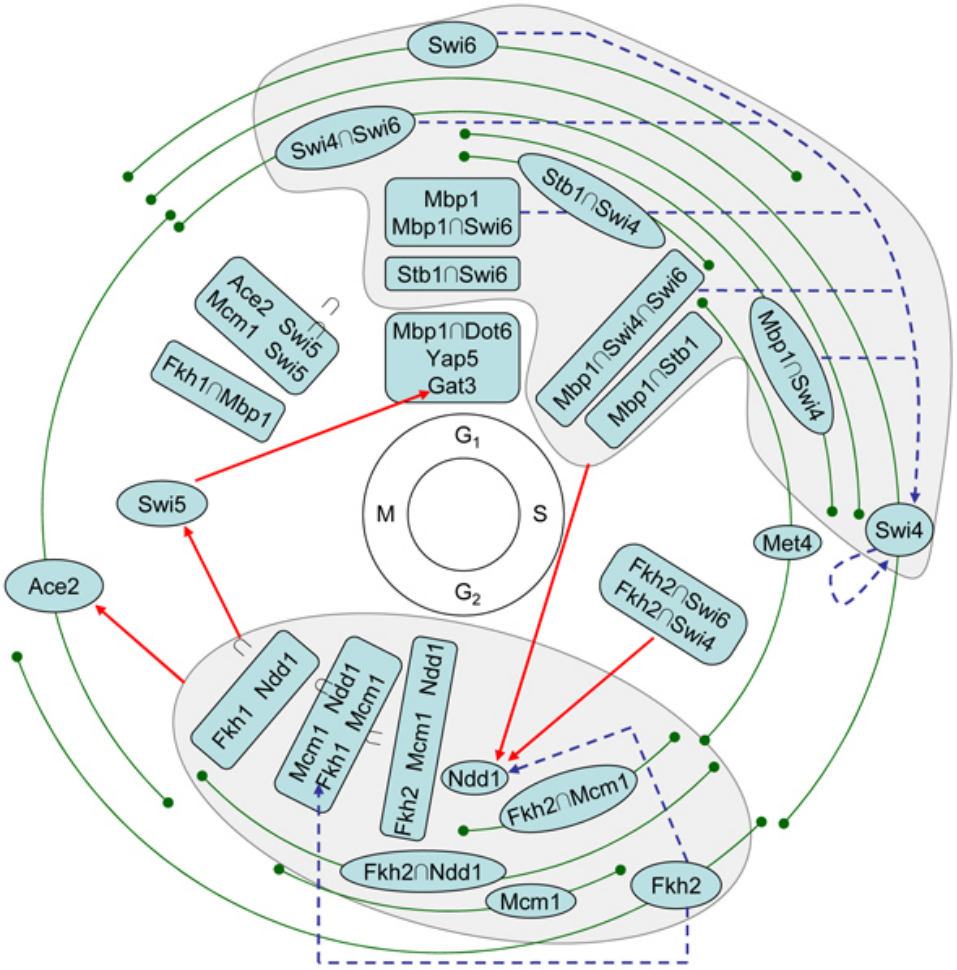
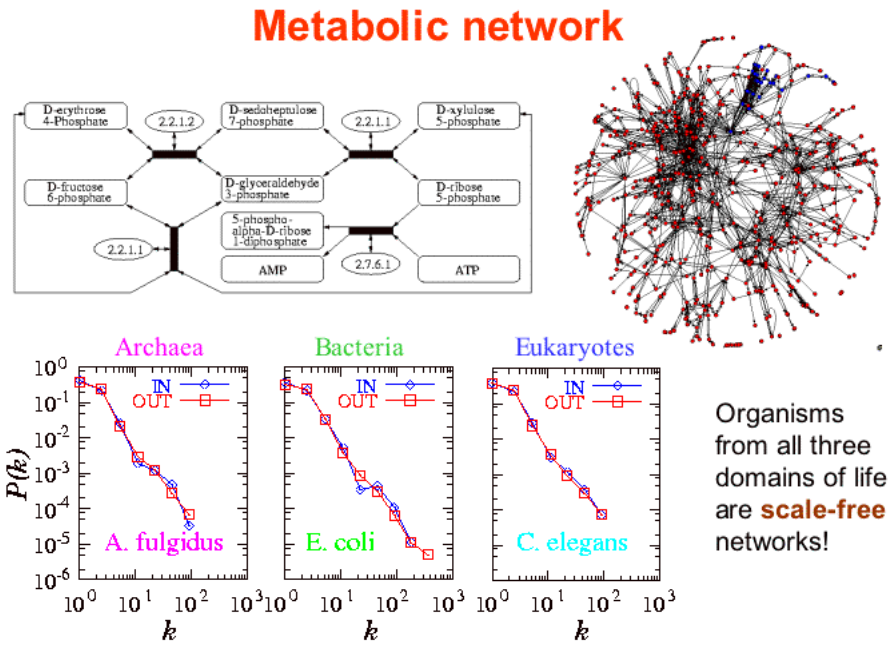


Figure 2.11: Transcription regulatory network in yeast. The figure is from [10].



H. Jeong, B. Tombor, R. Albert, Z.N. Oltvai, and A.L. Barabasi, Nature, **407** 651 (2000)

Figure 2.12: Example view for a metabolic network. The figure is from [11].

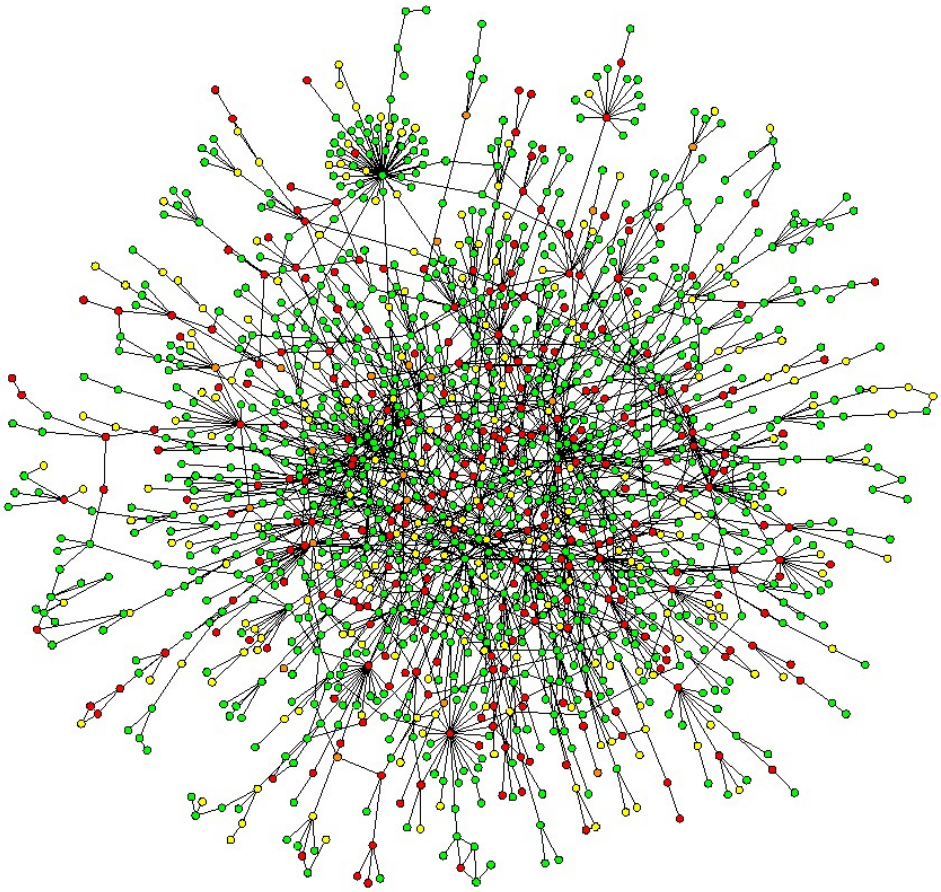


Figure 2.13: An example view of a protein-protein interaction network

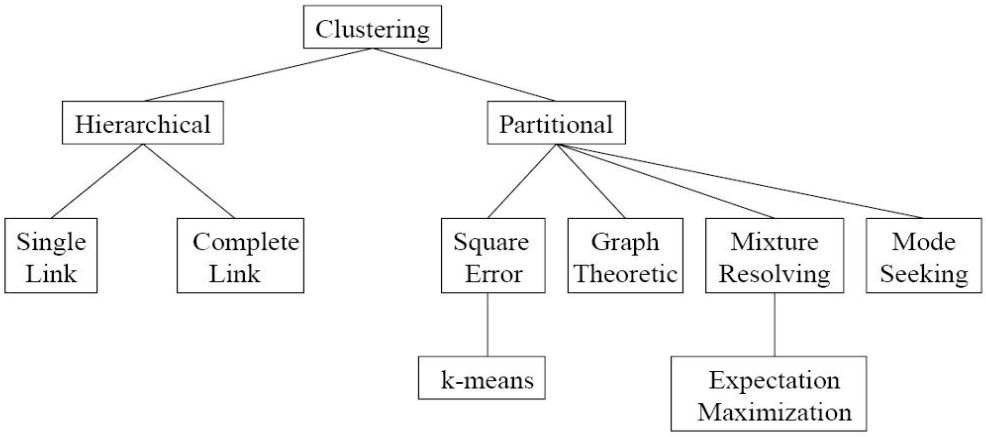


Figure 2.14: A taxonomy of clustering approaches, by the courtesy of [12].

Chapter 3

LARGE-SCALE OF PROTEIN MOTIF DISCOVERY WITH SNMF AND CHOU-FASMAN PARAMETERS

3.1 Background

Proteins are critical parts of organisms, providing structural or mechanical functions and participating in every process within cells such as cell signaling, immune responses, and the cell cycle. Proteins are complete biological molecules in a stable conformation that are made of twenty possible amino acids arranged in a linear chain. The chemical interactions of amino acid residues determine the conformation of proteins and form a relationship between protein sequences and structures. Therefore, understanding the close relationship between protein sequences and structures by discovering its hidden knowledge has been one of the primary interests in bioinformatics research.

A protein sequence motif is a recurring pattern in sequences that is prevalent in a number of proteins. Protein motifs are assumed to have biological significance such as binding sites and conserved domains. If a sequence motif is in the exon of a gene, it can encode a *structural motif* which is a three dimensional motif determining a unique element of the overall structure of a protein. With this property, sequence motifs can predict other proteins' structural or functional behaviors. Therefore, discovering sequence motifs is a key task to comprehend the connection of sequences with their structures.

PROSITE [45], PRINTS [89] and BLOCKS [90, 91] are currently the most popular motif databases. However, since the sequence motifs from these servers search through the same protein family members, they might carry little information about the consensus region beyond protein families [30]. On the other hand, many software programs for discovering one or more candidate motifs from a number of nucleotide or protein sequences have been developed. These include PhyloGibbs [92], CisModule [93], WeederH [94], and MEME [56]. For example, MEME utilizes

hidden Markov models (HMM) to generate statistical information for each candidate motif. However, such tools can handle only small to medium scale data sets and inappropriate for huge data sets.

3.2 Problem Statement

In order to obtain universally preserved sequence patterns across protein family boundaries, we use an extremely large data set collected from various protein families. After collecting a number of protein sequences, their protein family information is not used in further processing. Therefore, the task of discovering protein motifs is mainly divided into three steps: collecting all the possible protein segments with a fixed window size, clustering the segments, and evaluating the quality of discovered motifs with respect to its structural closeness. Collecting all the possible protein segments are completed in previous studies by Chen et al. [31, 32, 210, 211] using a sliding window technique from a protein profile data set. After clustering, evaluating the quality of discovered motifs is conducted by comparing the secondary structures in each cluster.

Therefore, clustering protein segments is the most challenging and crucial task. Previously, K-means clustering algorithms with supervised initial points were proposed by Zhong et al. [30] and Chen et al. [31, 32, 210, 211]. These methods improve on an earlier approach where naive K-means algorithm was used by Hand and Baker [212]. The improved K-means approach proposed in [30] increased the number of clusters having high structural homology, by selecting 'good' initial points from a number of preliminary results obtained by using a K-means algorithm with random initial seeds. Utilizing a granular computing strategy to divide the original data set into smaller subsets and introducing a greedier K-means algorithm [31,32], or subsequent filtering process with support vector machine [210,211], Chen *et al.* further improved the overall quality of the clusters in terms of biological, chemical, and computational meanings. Those high quality of motifs are used to predict local tertiary structure of proteins in [211] as well. However, these clustering techniques are undisciplined, insecure, and computationally expensive. They are actually supervised methods since they plug good initial cluster centers, which are evaluated and selected after several runs,

into a final K-means algorithm. Also, the selection process requires repeated runs of K-means and additional user setups, which increase the computational costs.

In this study, we propose to use sparse nonnegative matrix factorization (SNMF) [186, 213] to cluster the protein segments data set. Originally proposed as a dimension reduction method for nonnegative data, NMF has been successfully applied to several tasks in computational biology described by Devarajan [214]. Areas of applications include molecular pattern discovery, class prediction, functional analysis of genes, and biomedical informatics. As an extension of NMF, SNMF which imposes sparsity constraints on the low dimensional factors showed superior results for microarray data analysis with computational efficiency as demonstrated in [186]. Recently, Kim and Park [215] demonstrated that SNMF was able to produce more consistent results than K-means with random initial seeds, because SNMF tends to converge with any initial setups, while K-means algorithm is very sensitive to its initial setups. Additionally, we show how to incorporate a bio-statistics to improve the results with its high structural similarity. Unlike the previous methods, we avoid using the secondary structure of the data being studied in the process of clustering as the structure should be used only for evaluation. Instead, we use Chou-Fasman parameters which are statistical information on existing protein data which do not require knowing of the secondary structure of the proteins being studied.

3.3 Methods

Combining all the techniques aforementioned, the work conducts the following tasks. First, we use granular computing to split the extremely large collection of protein segments into smaller subsets and then use the Chou-Fasman parameters to add an analyzed structural information. SNMF is applied to each small subset in parallel. Our experimental results demonstrate that SNMF produces better results in terms of their structural agreements than other previous methods in [30–32].

3.3.1 Related Works

DNA or protein sequence motifs have been discovered through the studies of evolutionary conservation by de novo computation with various tools such as MEME [56], CisModule [93], PhyloGibbs [92] and WeederH [94]. However, these programs search protein motifs from a set of functionally related proteins, or from the same family members. Therefore, they are unequipped to discover patterns appearing across protein family boundaries. Expecting that protein motifs carrying biological significance can be found from different protein families as well, K-means clustering algorithms have been utilized for a large data set of proteins from diverse protein families in [30–32, 210, 211].

K-means clustering algorithms are efficient for large data sets, but performance is sensitive to initial points and the order of instances. Peña et al. [216] compared four different initialization methods for K-means algorithm: *Random, Forgy, MacQueen* and *Kaufman*. The random method initializes a partition of K clusters at random, while the Forgy method [217] randomly chooses K seeds as initial cluster centers and assigns each data to a cluster of the nearest seed. MacQueen [218] selects K random seeds but assignments follow an order of the seeds. The Kaufman method [219] successively picks K representative instances by choosing the center as the first one. According to the study in [216], Random and Kaufman methods outperform the other two methods, and the Kaufman method is faster than Random method. However, due to the stochastic nature of the large data used in the work of discovering motifs, Zhong et al. used the Forgy method as a traditional K–means in [30]. Throughout this thesis, the K-means with random initial seeds refers to the Forgy initialization strategy.

Previously, Han and Baker [212] utilized a K–means clustering with a random initial seeds to find protein motifs. Subsequently, Zhong et al. [30] introduced an improved K–means (Improved-K) algorithm that greedily chooses suitable initial centers so that final partition can reveal more protein motifs with structural similarity. However, good initial centers are selected from the resulting clusters obtained through previous K-means. Also this method requires two additional user inputs, a threshold for structural similarity hs, and a minimum distance between cluster centers md. That is, after a number of K-means, they select initial points having both produced the clus-

ters whose structural similarity are higher than hs and whose distance between already selected initial points in the previous run is farther than md. All the selected initial points are applied to the final K-means clustering algorithm. Although the Improved-K method was able to obtain more valuable clusters with higher structural homology (over 60%) than a traditional K-means algorithm, this method is actually supervised and led the results with manual selection of the cluster centers.

For further improvements, Chen et al. utilized granular computing introduced in [173, 174, 210, 211] and combined improved K-means or greedy K-means to develop the FIK model [31] and the FGK model [32, 210, 211], respectively. Fuzzy-Improved-K-means (FIK) model [31] and Fuzzy-Greedy-K-means (FGK) model [32, 210, 211]are granular based learning models used for the same task but for a larger data set than that of the improved K-means [30]. FIK and FGK both used Fuzzy C-means (FCM) algorithm for granular computing. FCM is a soft clustering algorithm which allows a data point to belong to one or more clusters [220,221]. FIK and FGK models divide the original data set into smaller subsets using FCM and slightly modified Improved-K [30] to each subset. While improved-K selected initial seeds sequentially, FIK collects all 'candidate' initial seeds from all of the preliminary K-means, then selects the ones which frequently appear and are reasonably distant from the other seeds. With FGK, the selection is more greedy by selecting the high quality of clusters first. However, although FIK and FGK models produced better results than the Improved-K, they are still manipulating the results by plugging good initial points into a final K-means run.

3.3.2 New Approaches

The previous models discussed in the previous section used K-means clustering algorithms with various initialization strategies. Instead of the K-means methods, we propose a different clustering algorithm called *sparse nonnegative matrix factorization (SNMF)*. The NMF and an SNMF algorithms are described in the Appendix A and readers are advised to refer the chapter. Here, we show how we utilized SNMF to discover large-scale of protein motifs.

3.3.2.1 Granular computing and Fuzzy C-means clustering

Chen et al. [31, 32] proposed a granular computing model [173, 174] by utilizing Fuzzy C-means (FCM) clustering algorithm. Granular computing involves the processing of complex information granules arising in the process of data abstraction and the derivation of knowledge from the data. FCM [220, 221], known as a common method for granular computing, is a clustering algorithm that allows a data point to belong to more than one clusters. Therefore, FCM is used as a preprocessing step as it splits the data with softer constraints. FCM clusters N data points, x_i's, into C clusters by minimizing the following objective function:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} \parallel x_i - c_j \parallel^2, \tag{3.1}$$

where m is the *fuzzification factor* and u_{ij} is the degree of participation of x_i into the cluster j with a center c_j. Then the number of clusters for each information granule divided by FCM is computed as,

$$C_k = \frac{n_k}{\sum_{i=1}^{m} n_i} \times \text{total number of clusters}, \tag{3.2}$$

where C_k is the number of clusters and n_k is the number of members for the k^{th} information granule.

FIK (Chen et al., 2006) [31] and FGK (Chen et al., 2006) [32] models applied FCM with empirically chosen fuzzification factor and the number of clusters, then applied K-means to each information granule with manually chosen initial points. They highlighted that not only the manual selection of initial points, but also the FCM process itself improved the final results due to its pre-filtering work as shown in Table 3.2.

In the present work, we apply SNMF method instead of variant K-means algorithms, because a sparse coefficient matrix can assign each data to one of the clusters. SNMF is known to be more consistent than K-means as it is converging to an optical point. However, less number of clusters can produce more desirable results with SNMF. If the number of clusters is numerous,

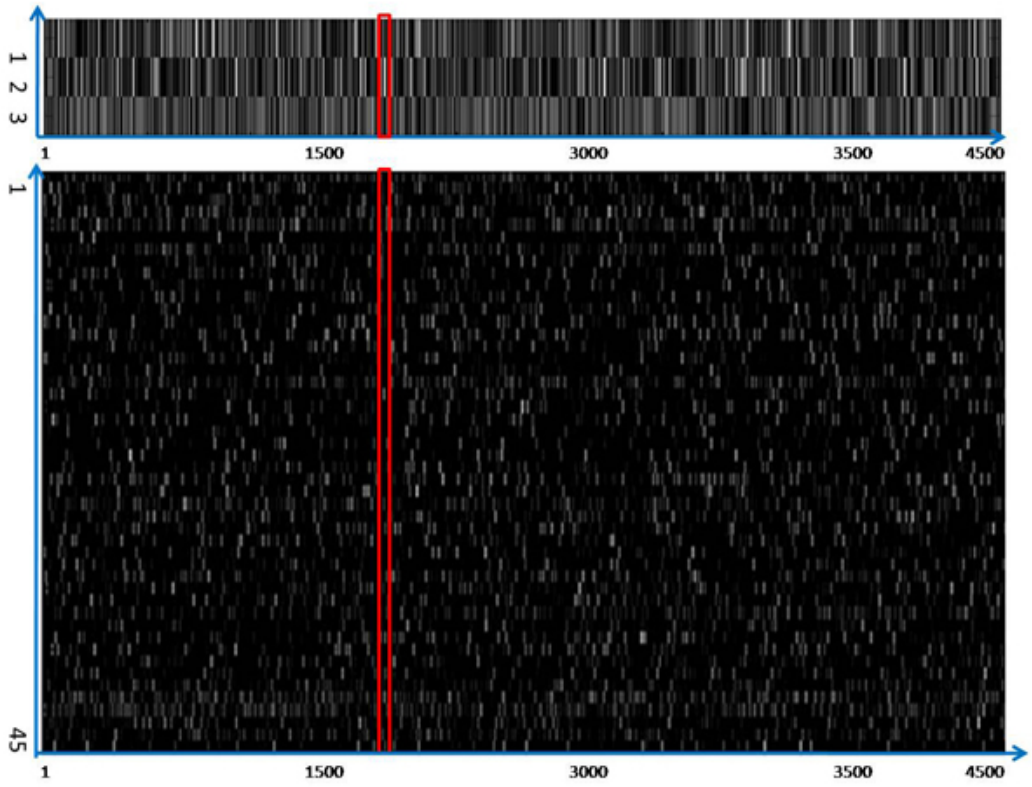


Figure 3.1: The top image is the coefficient matrix when $k = 3$ and bottom image is the coefficient matrix when $k = 45$. The y-axis represents the number of clusters and the x-axis is the data point. For a specific data shown as a red vertical box, the assignment of the top matrix is clearer than the bottom matrix, as the second row clearly beats the others. The bottom coefficient matrix has more than 7 non-zero values holding around 10% of the weight each, making a proper assignment difficult.

then many of the relevant factors hold similar weights, thereby obstructing proper assignments. Figure 3.1 shows one example of obscure assignments by comparing with the case of $k = 3$ and that of $k = 45$. The bottom image of Figure 3.1 visualizes the H factor with 45 rows. One data shown within a red box has 45 weights, but not one value is promising enough to assign the data to a particular set. The top image of Figure 3.1 only has 3 rows and the second value is prominent enough to cluster the data of red box into the second set. Hence, the granular computing with FCM is a crucial step for clustering with SNMF. Instead of one FCM to divide the data set, we applied FCM hierarchically to avoid data over-fitting. We carefully picked the proper fuzzification factor through experiments and strictly enforced the amount of data overlapping for this double FCM

process. As a result, we improved the final results in terms of the structural homology and reduced the overall spatial and temporal complexities as well.

3.3.2.2 Chou-Fasman parameters

K-means algorithm is known to considerably depend on initial centers, which can lead to a poor local optimal solution rather than the global optimal one. Therefore, Zhong et al. [30], and Chen et al. [31, 32, 210, 211] chose 'favorable' initial points to plug into a K-means and increased the number of clusters with high structural similarity. However, the selection of good initial points involve knowing the results in advance. That is, a number of executions of K-means algorithm preceded and the resulting clusters are evaluated with its secondary structure similarity. The selection of favorable initial points from the 'good' clusters is then followed. This process is actually a supervised learning method which is undesirable for clustering.

Therefore, we use SNMF to cluster the data set without supervising the procedure. SNMF is proven to be more consistent than K-means in the study [215], meaning that with any initial points the results tend to converge closely to a global optimal point. In the experiment, we actually observed that the primary sequence groupings is much better with SNMF than K-means with initial random seeds. Computationally, however, the resemblance of primary sequence does not guarantee the similarity of secondary structure in a cluster. To infer the clusters of high structural homology from its primary sequence, we used Chou-Fasman parameters to add a statistical relationship between primary sequence and secondary structure into the data set.

Chou-Fasman parameters shown in Table 3.1 were first introduced by Chou and Fasman [222, 223]. Each amino acid is assigned to ***conformational parameters*** of $P(a), P(b)$ and $P(t)$, which represent the tendency of each amino acid to alpha helices, beta sheets and beta turns, respectively. The parameters were determined by observing a set of sample protein sequences of known secondary structure. The additional parameters of $f(i)$, $f(i+1)$, $f(i+2)$ and $f(i+3)$ correspond to the frequency with which each amino acid was examined in the first, second, third or fourth position of a hairpin turn. For additional structural information, we compute the tendency of secondary structures based on the frequencies of amino-acid residues at each location, with the

three conformational parameters of $P(a), P(b)$ and $P(t)$. For example, if a location at a protein segment consists of 20% of Alanine and 80% of Cysteine, then the relevant secondary structure for the location is 28% of alpha helices, 37% of beta sheets and 35% of beta turns. The statistical structural information is included into the data set. Details are described in Section 3.4.1.

3.3.3 Evaluation Methods

We emphasize that this is an unsupervised learning task, meaning that no prior information about data grouping is given. Hence, after we cluster the data set into similar protein groups, we need biological measures to evaluate the clusters to discover more qualified motifs. Zhong et al. [30] suggested a measure of secondary structure similarities in order to capture close relationships between protein sequences and their structure, and Chen et al. [31, 32] additionally proposed a biochemical measure, HSSP-BLOSUM62, as well as a computational measure of the David-Bouldin Index (DBI) measure. In this thesis, we use the secondary structure similarity evaluation which is used in common in the previous studies [30–32], and additionally introduce a new evaluation measure, called sDBI, which is the DBI measure for the computed secondary structure.

Secondary Structure Similarity measure The structural similarity of each cluster is computed as the following:

$$\frac{\sum_{i=1}^{ws} max(P_{i,H}, P_{i,E}, P_{i,C})}{ws}, \tag{3.3}$$

where ws is a window size and $P_{i,H}$ is the frequency of the helix at the i^{th} position of the segments in the cluster. $P_{i,E}, P_{i,C}$ are defined similarly for beta sheets and turns. After a clustering, each cluster is evaluated with its secondary structure similarity, and clusters with more than 60% similarity are counted, since proteins exceeding 60% structural homology are considered structurally similar [30, 224]. A method producing more clusters with over 60% structural homology will be considered better method with this measure.

Structural David-Bouldin Index (sDBI) measure Besides the biological measure of secondary structure similarity, Chen et al. [32] used a computational evaluation called David-

Bouldin Index (DBI) measure [225], to evaluate the groupings only in terms of their primary sequence. The DBI measure is a function of intra-cluster (within-cluster) distance and inter-cluster (between-cluster) distance. Because a cluster with a relatively larger inter-cluster distance and a relatively smaller intra-cluster distance is more favorable, a lower DBI indicates a better data groupings. Equation (3.4) computes the DBI value of a clustering task.

$$DBI = \frac{1}{k}\sum_{p=1}^{k} max_{p\neq q}\{\frac{d_{intra}(C_p) + d_{intra}(C_q)}{d_{inter}(C_p, C_q)}\}, \tag{3.4}$$

where $d_{intra}(C_p)$ is the average of all pairwise distances between each member in the cluster C_p and its center, and $d_{inter}(C_p, C_q)$ is the distance of the centers of two cluster C_p and C_q, and k is the number of clusters. All the distance is computed in Hamming distance metric.

However, the DBI of a primary sequence evaluates grouping behavior only in terms of primary sequences. In fact, if we compare the performances based on DBI, we could show that SNMF produce better results than a K-means with random initialization. But, DBI measure is improper for finding qualified motifs since good clusters in terms of primary sequences have little biological significance. Therefore, we introduce a new measure which evaluates its computational clustering results based on the inferred structural information. We call this new measure *Structural David-Bouldin Index measure* (sDBI). The sDBI follows the same equation as DBI in Equation (3.4). The difference is that each cluster consists of the inferred secondary structure S instead of the primary sequence O in Equation (3.5). By using sDBI, we can evaluate the overall grouping qualities not restricted to finding a subset of good clusters.

3.4 Result and Discussion

The work uses the same data in [31, 32, 210, 211], which extended the data used in [30]. By reviewing detailed description of data representation and their measure of previous studies of [30–32, 210, 211], we design a new measure which can evaluate the quality of overall clusters. The performances of each method will be compared with two measures.

3.4.1 Data set and data representation

A total of $2,710$ protein sequences, none of which shares more than a 25% sequence identity, from Protein Sequence Culling Server [226] are used in this work. By sliding a window of size 9 through each sequence, we collect more than $560,000$ sequence segments. Each segment is represented as the frequency profile constructed from HSSP [224], based on the aligned sequences from protein data bank (PDB). The secondary structure of each segment, which will be used to evaluate the results, is also obtained by DSSP [227]. Hence, as shown in Figure 3.2, each primary sequence segment of length 9 forms a 20×9 matrix, where each location has the frequencies of 20 amino acid residues in the vertical direction.

In this study, we apply FCM to primary sequences, then we add secondary structure statistics inferred by Chou-Fasman parameters to the original data format before applying SNMF. The additional data structure is computed as follows. Let S to be the statistically inferred secondary structure of a 3×9 matrix format with a length of 9 and three types of secondary structure for the helices, beta sheets and beta turns. Let O be the original sequential data shown in Figure 3.2, with a 20×9 matrix format. Since $O(i, j)$ represents the frequency of the i^{th} amino acid at the j^{th} location, we want $S(i, j)$ to stand for the probability of the i^{th} second structure at the j^{th} location. We obtain a 3×20 matrix for the Chou-Fasman parameter C, where $C(i, j)$ is the percentage of i^{th} structure for j^{th} amino-acid. Then S is computed as

$$S = C \times O. \tag{3.5}$$

The final data format for SNMF is the combination of the primary sequence, O, and the computed secondary information, S, forming a 23×9 matrix. That is, each position includes the frequencies of H, E, and C in addition to the frequencies of 20 amino-acid residues.

Finally, each data is unfolded into a $23 \times 9 = 207$ dimensional vector and n number of data are formed into $207 \times n$ data matrix A for SNMF. The sparse factor H now directs an assignment of each data sample to one of k clusters.

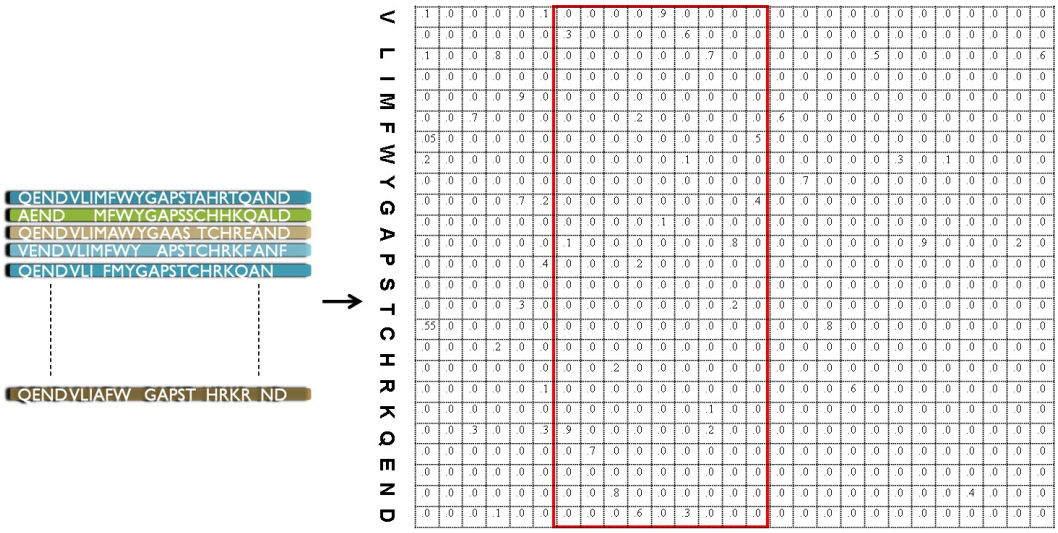


Figure 3.2: A number of protein sequences in a protein family obtained from PDB server are aligned on the left. According to the frequencies of twenty amino acids represented as one-letter codes, the proteins are expressed as a profile data on the right figure. Sliding a window of length 9, the 20×9 matrix shown inside the red box represents one protein segment data format.

3.4.2 Experiment steps

Protein motifs discovery using SNMF method follows the subsequent steps. We divide the data set into a number of small size of subsets using FCM hierarchically, for proper clustering task with SNMF method. We first split the data set into ten smaller subsets using FCM, then divide each subset further into much smaller subsets with another FCM. Although the 'FCM + SNMF' model, shown in Table 3.2, increases the percentage of structurally significant clusters, we utilize the conformational parameters of Chou-Fasman table to compute the structural relationship with primary sequence, to improve the results further. Figure 3.3 summarizes the experiment steps conducted in this study. To see the impact of Chou-Fasman parameters, we applied a K-means with initial random seeds to the combined data set, and provided the result as well.

3.4.3 Clustering Results

We summarize the clustering results in Table 3.2. Each method is compared with two measures: the secondary structure similarity and the sDBI. The first column indicates the methods we used in this study as well as Improved-K by Zhong et al. [30], FCM, FIK and FGK by Chen et

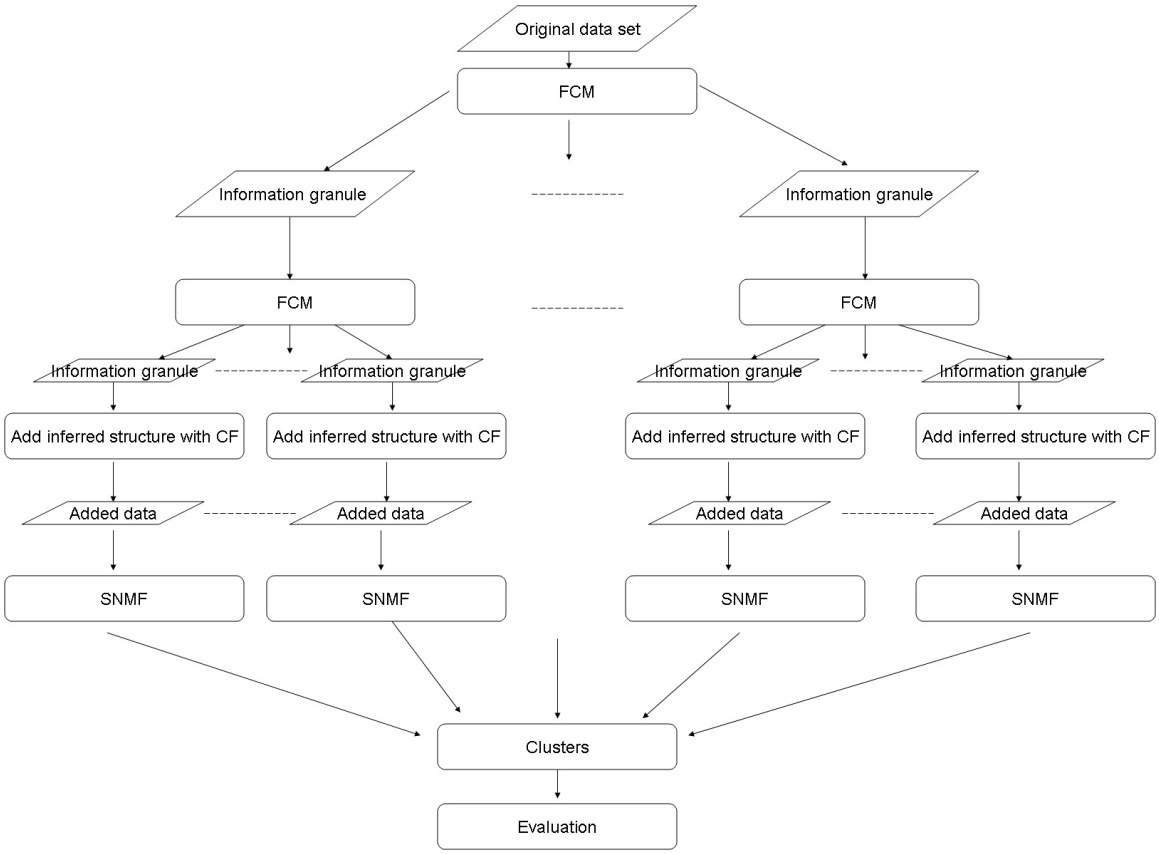


Figure 3.3: This figure summarizes the experiment steps in this study. The original data set of a primary sequence is divided into smaller subsets (information granules) with double applications of FCM. For each subset, secondary structure information is inferred with Chou-Fasman parameters and added to each data set. SNMF is finally applied to each subset and the result is evaluated using two evaluation criteria, secondary structure similarity and sDBI.

al. [31,32]. The second column is the percentage of clusters which have a secondary structure similarity exceeding 60%. The next column is the percentage of clusters having a structural similarity greater than 70%. For structural similarity, a higher percentage is more favorable. The last column indicates sDBI value of each method. With sDBI values, lower values are preferred. The first five methods listed in Table 3.2 are from Zhong et al. (2005) [30] and Chen et al. (2006) [31, 32], and they are provided to be compared with our models. We excluded the results by Chen et al. (2008) [210, 211] since the studies added further filtering procedure after clustering.

'Traditional' is a K-means with random initial seeds and 'Improved-K' is the method by Zhong et al. (2005) [30]. 'FCM' is granular computing combined with a K-means with random

initial seeds. With FCM, the increased percentage of good clusters having high secondary structure similarity shows a significant improvement over a traditional K-means. FIK (Chen et al. (2006) in the fourth row in Table 3.2 illustrates further improvement, and FGK (Chen et al. 2006) model produced the best result among all of the previous models. The sDBI is a new measure introduced in this thesis, and we were unable to provide sDBI values for previous models except FGK, since the resulting clusters of other models were unavailable. Reproduction of the results were also impossible as these results are obtained through lots of experimental trials with different settings. As the result with FGK was obtainable from the authors, we could compute sDBI of FGK result only.

The rest of Table 3.2 lists some of the experiments we conducted in this study. 'FCM+SNMF' shows the result of applying FCM followed by an SNMF, without the Chou-Fasman parameters. The structural homology indicates that SNMF provides more qualified motifs than other results with structural similarity measure. However, it did not beat sDBI value of 'FGK' model, requiring another way to improve the clustering result further. Therefore we incorporated secondary structure information computed with Chou-Fasman parameters into the data set and were able to see an improvement on both measures. To see the influence of Chou-Fasman parameters on K-means, we applied this incorporated data to K-means with random initial seeds too, and the 'FCM + CF + K-means' model shows further improvement than FCM, in terms of the structural homology. Since these models are using random initial seeds, we can expect to have further improvement when using greedy K-means algorithm on this combined data.

Finally, we further improved both the structural and the computational qualities using the 'FCM+CF+SNMF' method (Kim et al. 2011) [77] shown in the last row of Table 3.2. As summarized in Figure 3.3, we divided the original data set into much smaller information granules by applying FCM hierarchically, then added secondary structure statistics inferred by Chou-Fasman parameters and primary sequences. Then, we processed SNMF to obtain a sparse coefficient factor for clustering. As a result, the last model bettered the performance of the previous best model, 'FGK', for both the structural similarity and sDBI measures. In conclusion, the 'FCM+CF+SNMF' demonstrates that the combination of extended data with structure statistics along with SNMF can

discover more structurally meaningful motifs. The result is actually proving that using SNMF, we can obtain more qualified motifs with proper unsupervised clustering method, without manual setting of cluster centers.

3.4.4 Sequence Motifs

Figure 3.4 3.5, 3.6, 3.7, 3.8 are five different sequence motif examples discovered in this study. They were created using the Weblogo tool [48]. Weblogo is a web-based tool that generates sequence logos which are graphical depictions of the sequence patterns within a multiple sequence alignment. We illustrate some of the motifs found by our method with sequence logos as they provide a richer and more precise description of sequence similarities than consensus sequences or the previous formats used in [30–32]. The sequence logos are obtained from the clusters which have over 60% secondary structural similarity, and more than $1,000$ protein segments. The exact number of segments and the structural homology are given at the top of each motif image. The motif pattern is represented starting from the N-terminal and the letters stacked at each position demonstrate the type of amino acid which appears with over 8% frequencies in that position. The height of symbols indicates the relative frequency. The letter shown below the x-axis is the representative secondary structure of that position, where H is for helix, E for sheets and C for turns. For example, Figure 3.4 is a motif of helix-structure with conserved Alanine (A), and Figure 3.6 is a turn-sheet motif and its second position consists of four amino acid (D,G,E,S) with roughly equal frequencies.

3.5 Summary and Future Work

In this study, sparse nonnegative matrix factorization (SNMF) combined with granular computing and inclusion of statistical structure is proposed to discover protein motifs which are universally conserved across protein family boundaries. Discovering high quality of protein motifs is very useful in the study of bioinformatics, as the sequence motifs can reveal structural or functional patterns. For example, Chen et al. [211] showed that the sequence motifs can be used to predict protein local tertiary structure. Previous models proposed in [30–32, 210, 211] involve K-means

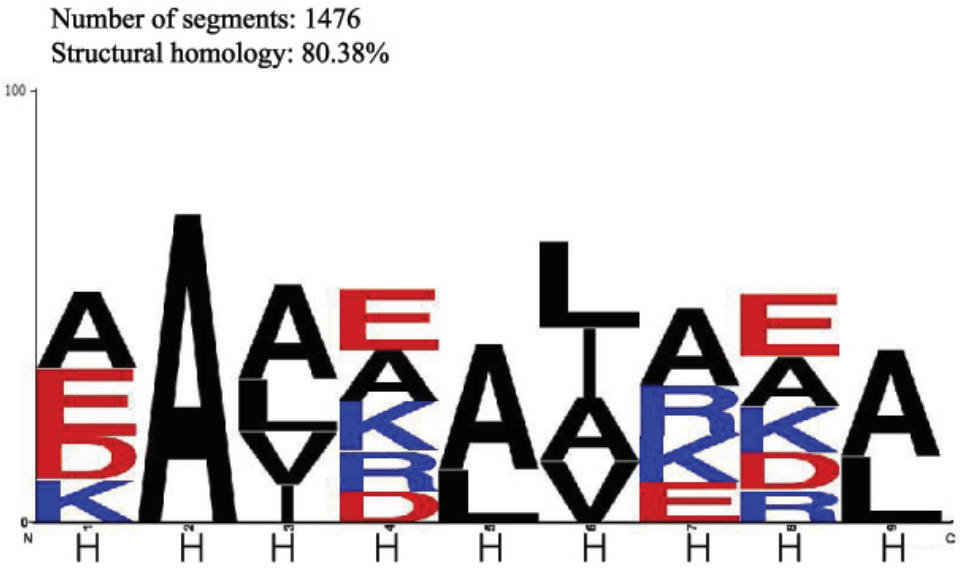


Figure 3.4: Helices motif with conserved A

clustering algorithms with various initialization strategies. However, in the process of initialization, they used the secondary structure of the data being examined which should be used only for evaluating the results. Therefore, the previous models are undesirable as they are actually supervised clustering methods. Instead, we use an SNMF clustering method since it is more consistent and efficient than K-means algorithms with manually selected initial points. In addition, we found that the incorporation of Chou-Fasman parameters plays an important role for this task. Besides the secondary structure similarity measure, which is limited to selecting a subset of good clusters, we designed a new measure, sDBI, which evaluates the overall grouping qualities based on the inferred secondary structures and the primary sequences. We also observed that the process with SNMF is less expensive and more meaningful if the size of each subset is reduced with Fuzzy C-means preprocessing.

The work makes four contributions in the study of molecular biology. First, we explore the use of SNMF to a new problem domain, protein profiles. NMF has been used for various data including image, text, microarray gene or protein expression data. As far as we know, this is the first time that NMF has ever been applied to a protein profile data set. Even with the same SNMF algorithm, adjustment of parameters to different data format was a challenge. Second, we adopt

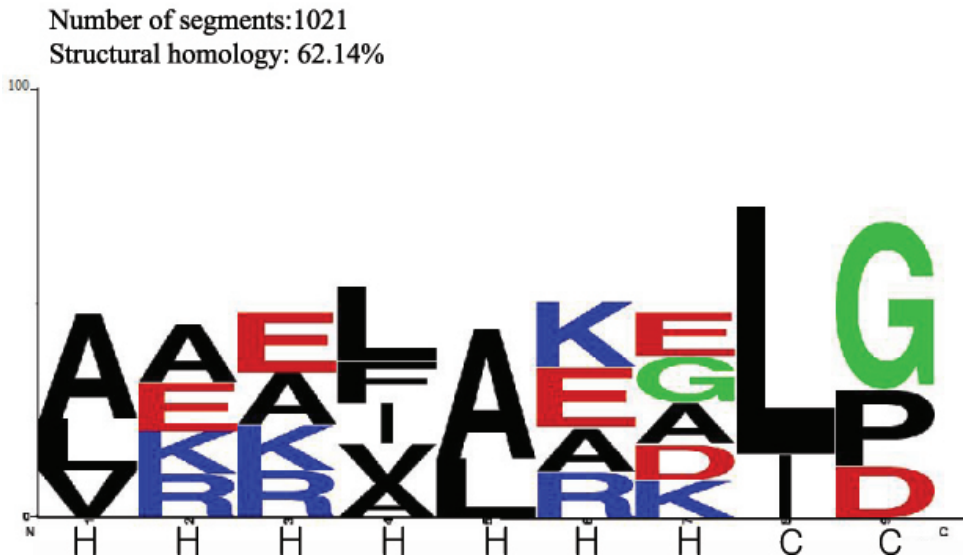


Figure 3.5: Helix-Turn motif

Chou-Fasman parameters [222, 223] which give us the statistical relationship between sequence and secondary structure and improve the quality of resulting motifs. It is also shown that the inclusion of Chou-Fasman parameters itself is a powerful tool to improve the quality of clusters even with a K-means algorithm with random initial seeds. Third, by applying granular computing strategy, we were able to overcome the issues with obscure assignments with SNMF method for large data sets. The final contribution is designing a new measure which evaluates the quality of motifs based on a 'statistical' structural information inferred from its primary sequence. With this measure, we can evaluate its structural significance without loss of sequential similarity.

Sparse nonnegative matrix factorization method, however, does have its limitations. For better clustering results, the number of clusters should be small. Otherwise, the presence of many nonzero coefficients holding similar weights make the assignment task obscure. Therefore, an additional dividing process is required, which in turns increases computational complexities and risks of data overfitting. In addition, as with K-means clustering, the number of clusters need to be determined as a prior parameter for the SNMF method, hindering an automated optimization.

Therefore, our future works include the followings. We need to find a way to decide an optimal number of clusters automatically. Resolving the problem of assigning data to a cluster when there are one or more candidates is another area of future interest. It is also necessary to

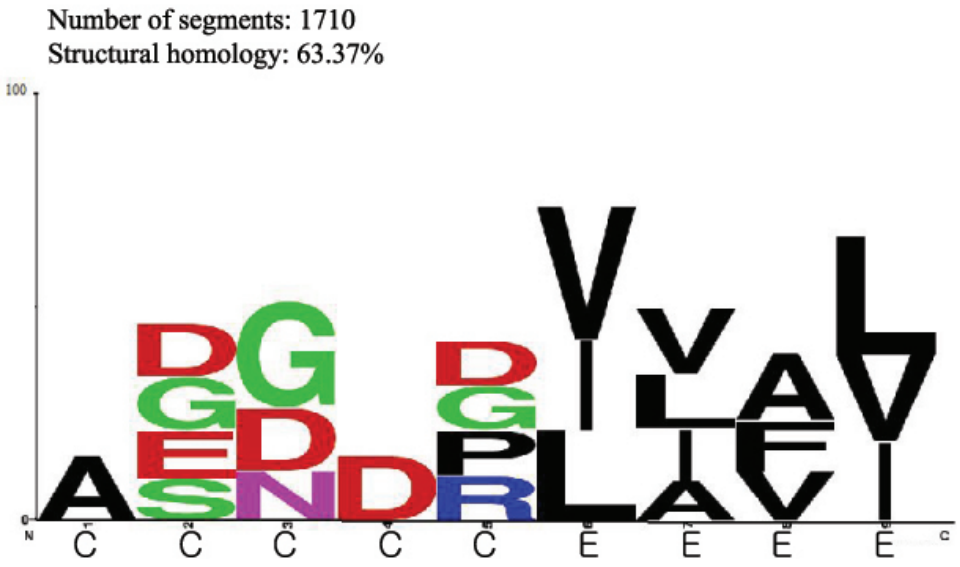


Figure 3.6: Turn-Sheet motif

reduce the computational costs and risks caused by additional dividing steps. We also want to add other evaluation methods, such as functional homology, to qualify the discovered motifs. Finding more biological applications with the protein motifs discovered through this study would be very important future study as well.

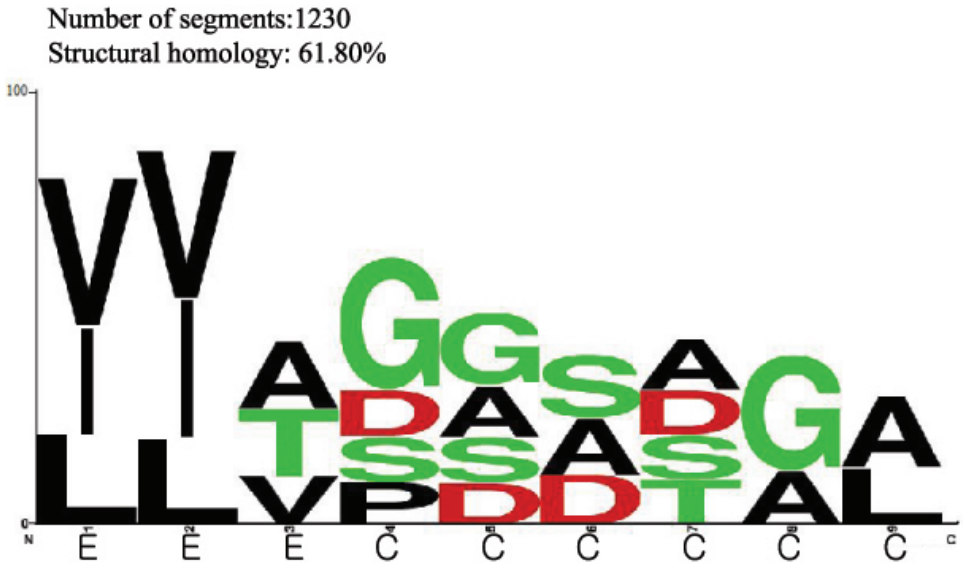


Figure 3.7: Sheet-Turn motif

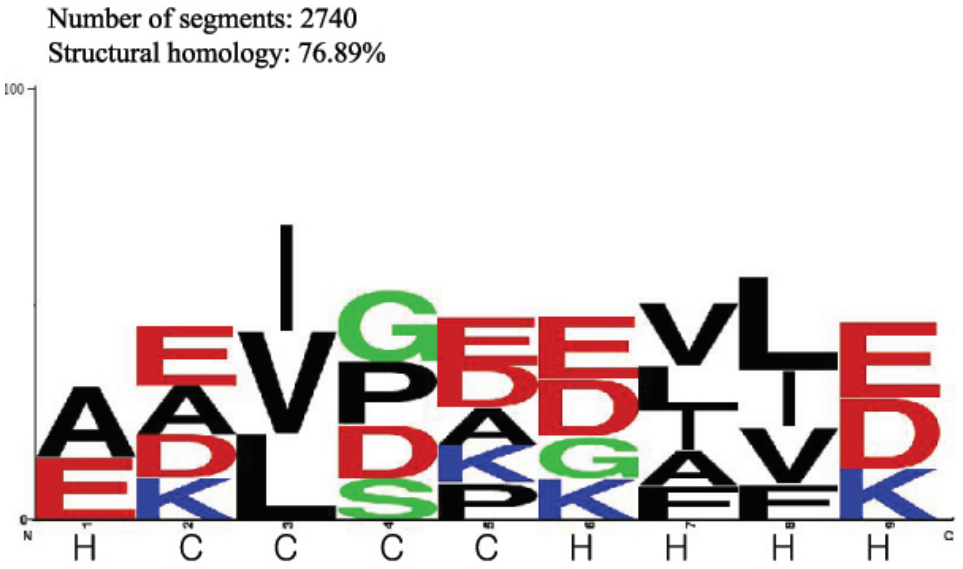


Figure 3.8: Helix-Turn-Helix motif

Table 3.1: Chou-Fasman parameter

Symbol and name of Amino Acid	P(a)	P(b)	P(t)	f(i)	f(i+1)	f(i+2)	f(i+3)
A : Alanine	142	83	66	0.66	0.076	0.035	0.058
R : Arginine	8	93	95	0.07	0.106	0.099	0.085
D : Aspartic Acid	101	54	146	0.147	0.110	0.179	0.081
N : Asparagine	67	89	156	0.161	0.083	0.191	0.091
C : Cysteine	70	119	119	0.149	0.050	0.117	0.128
E : Glutamic Acid	151	37	74	0.056	0.06	0.077	0.064
Q : Glutamine	111	110	98	0.074	0.098	0.037	0.098
G : Glycine	57	75	156	0.102	0.085	0.19	0.152
H : Histidine	100	87	95	0.14	0.047	0.093	0.054
I : Isoleucine	108	160	47	0.043	0.034	0.013	0.056
L : Leucine	121	130	59	0.061	0.025	0.036	0.07
K : Lysine	114	74	101	0.055	0.115	0.072	0.095
M : Methionine	145	105	60	0.068	0.082	0.014	0.055
F : Phenylalanine	113	138	60	0.059	0.041	0.065	0.065
P : Proline	57	55	152	0.102	0.301	0.034	0.068
S : Serine	77	75	143	0.12	0.139	0.125	0.106
T : Threonine	83	119	96	0.086	0.108	0.065	0.079
W : Tryptophan	108	137	96	0.077	0.013	0.064	0.167
Y : Tyrosine	69	147	114	0.082	0.065	0.114	0.125
V : Valine	106	170	50	0.062	0.048	0.028	0.053

The first column is the name of twenty amino acids with its corresponding one-letter code in parentheses. The next three columns represent the propensities of each amino acid for α-helices $(P(a))$, β-sheets $(P(b))$ or turns $(P(t))$. The rest of the parameters $f(i+j)$'s are the tendencies of the $j+1^{\text{th}}$ position of a hairpin turn, which are generally used to predict a bend.

Table 3.2: Comparison of various clustering methods

Methods	$> \mathbf{60}\%$	$> \mathbf{70}\%$	**sDBI**
Traditional K-means	25.82%	10.44%	N/A
Improved-K (Zhong et al. 2005, [30])	31.62%	11.50%	N/A
FCM (Chen et al. 2006, [31])	37.14%	12.99%	N/A
FIK (Chen et al. 2006, [31])	39.42%	13.27%	N/A
FGK (Chen et al. 2006, [32])	42.93%	14.39%	7.21
FCM+CF+K-means	42.94%	13.23%	9.07
FCM+ SNMF	44.07%	12.73%	9.85
FCM + CF + SNMF (Kim et al. 2011, [77])	$\mathbf{48.44}\%$	$\mathbf{16.23}\%$	$\mathbf{7.05}$

The first five rows summarize the results of previous methods introduced in [30–32]. The rest of methods list the experiments conducted in this study. The last result is the best result obtained for both measures. This result was obtained by using an SNMF which was combined with FCM and Chou-Fasman parameters.

Chapter 4

ALGORITHMS AND EVALUATION METHODS FOR BIOLOGICAL NETWORK MOTIF

4.1 Background

Systems biology focuses on the study of complex interactions in biological systems, rather than the study of individual molecules such as DNA, RNA, proteins and metabolites [118]. One of the goals of systems biology is understanding the structures of all molecules and their interactions in a system level. Therefore major challenges include expressing the dynamic structures of small molecules and determining their functions in a living cell. Various types of biological interactions have been expressed in networks, such as, transcriptional regulatory networks, signaling pathways, metabolic networks and protein-protein interaction (PPI) networks. Biological networks share some of structural properties of other complex networks, or have specific features of scale-free and small-world effect [228]. However, the properties have been questioned by Lacroix et al. [159] with a number of reasons including the incompleteness of networks and inconsistent link generations of the graphs. Therefore, the analysis extends to other network properties such as network clusters and network motifs.

As biological networks are massive and the size is still increasing, dividing the network into a number of clusters helps reveal specific local properties. Network motif, as another concept describing local properties of a network, is defined as a small connected subgraph appearing frequently and uniquely in a network. Similar to a protein sequence motif, a network motif is defined as an over-repeated pattern, but detection of it requires much more computation as the process involves isomorphic testing and repeated processes for uniqueness determination. Network alignment [6] and network querying [7] are also local analyses of networks, but while network motifs are studied with only structural information, network alignment and network querying are studied with both of the topological and biological properties.

Previous network motif discovery algorithms include exact counting and approximation algorithms: Exhaustive recursive search (ERS) [16], enumerate subgraphs (ESU) [15] and compact topological motifs [17] are exact counting algorithms. Exact counting algorithms face extreme computational challenges if the network size is large or the motif size is large as shown Figure 2.8 and Figure 2.7. The blue line in Figure 2.8 shows that the computational time increases rapidly as the size of network, that is, the number of edges linearly increases. The dashed line is a polynomial graph of order two as a trend line to show its trend. Also the blue line in Figure 2.7 shows that the search time increases exponentially as the size of motifs increases, with the exponential trend graph of dashed line. From these trends, therefore, exact counting algorithms are infeasible as most of biological networks are huge. Consequently, several approximation algorithms have been provided including edge sampling (MFINDER) [16], randomized version of ESU from a search tree (RAND-ESU) [18], and tree-filtering search which is NEMOFINDER [19]. However, most of approximation algorithms do not guarantee correct results. Hence, various parallel algorithms have been introduced to realize feasible exact counting algorithms [20, 21].

Network motifs are used for many applications in biological networks. Feed-forward-loop (FFL) and Bifan network motifs are identified as the typical patterns in different types of biological networks [101, 229]. Przulj et al. [23] used network motifs as a relative graphlet frequency distance to distinguish different protein-protein interaction networks. Also motif frequencies are exploited as classifiers for network model selection [24]. Milo et al. [25] studied that networks of different biological and technological domains have been classified into different superfamilies on the basis of motif significance profiles. To predict protein-protein interactions, Albert I. and Albert R. [26] used network motifs successfully. In the study by Conant and Wagner [27], network motifs in transcriptional regulatory networks are not evolutionary conserved while network motifs in PPI networks are evolutionary related. On the other hand, network motifs are extended to ‘motif modes’ each of which has a certain topology and a specific functional property [28].

4.2 Problem Statement

Through a number of network motif applications, we notice several problems regarding the biological meanings of network motifs, on top of the computational challenge. First, the biological quality of network motifs are not validated thoroughly. A network motif is selected only by its structural uniqueness and they are mostly used for identifying some structural property in networks. Second, while there are databases of sequence motifs [45, 89] where the motifs are categorized according to some biological functions, there are no databases of network motifs. Third, non-motifs, that is, structurally insignificant subgraphs, have not been analyzed in any studies, which are filtered out before applied to any applications. Fourth, it is still questionable what the network motifs really represent in biological networks, whereas sequence motifs are known to have some biological functions.

Therefore, we want to focus more on the biological quality of network motifs, but still save the computational resources in an efficient way. We pursuit to develop innovative algorithms that search biologically useful motifs in an early stage so that minimize wastes afterwards. To see the usefulness of non-motifs in biological applications, non-motifs are also analyzed with their relationships with biological functions or protein complexes. Through these steps, we hope to find some representative biological functions of network motifs so that help constructing databases of network motifs for further usages. Although the work in this chapter is yet an initial step, we hope we provide some guidelines for the study of network motifs in biological contexts.

4.3 Methods

We first define ***biological network motifs*** as we want to focus more on biological meanings of network motifs. And we refer conventional network motifs as ***structural network motifs*** to distinguish them from biological network motifs. Unlike structural network motifs, biological network motifs are biologically significant small connected subgraphs. Biological significance of biological network motifs is unspecified in the definition, as it will be assigned flexibly according

to the application, such as, biological modules, elements of protein complexes or evolutionary relations.

For efficient detection of biological network motifs, we introduce EDGEBETWEENNESS-BNM, EDGEGO-BNM,NMF-BNM, NMFGO-BNM and VOLTAGE-BNM algorithms, and design new evaluation measures named, 'motifs included in complex', 'motifs included in functional module' and 'GO term clustering score' to validate the motifs. Our algorithms compete with existing algorithms including ESU, RAND-ESU and MFINDER, and the performance are compared based on the new evaluation measures aforementioned. The main idea for the algorithms is to reduce the search time by removing a number of edges from the original network and, at the same time, increase the discovery rate for biological network motifs. Experimental results with a couple of *S. cerevisiae* PPI networks demonstrate that EDGEGO-BNM and EDGEBETWEENNESS-BNM algorithms perform better than other algorithms overall. In addition, we show that all of our algorithms efficiently search for structural network motifs as well, so they can be alternatives of approximation motif finding algorithms, such as, RAND-ESU and MFINDER.

4.3.1 Definitions

We assume that a biological network is a graph $G = (V, E)$ where each vertex in V is a molecule and each edge in E is an interaction between vertices. A **network motif** m is an overly represented connected subgraph of size k in a graph, as defined in the Definition 4.3.1. The size of network motif, k, ranges from 3 up to 15 or more, but relatively very smaller than the size of the whole network, $|V|$.

Definition Let $G = (V, E)$ be a graph, and $3 \leq k << |V|$. A **network motif** m is a connected subgraph of size k in G, which appears more frequently than usual.

To determine the uniqueness of m, a number of random graphs, typically more than $10,000$ graphs, are generated and the frequency of each random graph R, which is $f_R(m)$, is recorded to

obtain P-value as in Equation (4.1) or Z-score in Equation (4.2).

$$\text{P-value}(m) = \frac{1}{N}\sum_{n=1}^{N} c(n), \text{where } c(n) = \begin{cases} 1, \text{if } f_R(m) \geq f_G(m). \\ 0, \text{otherwise.} \end{cases} \tag{4.1}$$

$$\text{Z-score}(m) = \frac{f_G(m) - average(f_R(m))}{std(f_R(m))} \tag{4.2}$$

Here, $f_G(m)$ and $f_R(m)$ are the frequencies of m in the target graph G and a random graph R, respectively. N is the number of random graphs and $average(f_R(m))$ and $std(f_R(m))$ refer to the average and standard deviation of frequencies in random graphs, correspondingly. Generally, a subgraph with P-value less than 0.01 or Z-score greater than 2.0 is considered as a network motif.

We define a **biological network motif** g as a small connected subgraph of size k which has topological property as well as biological meanings as in Definition 4.3.1. Note that we do not provide what 'biological significance' specifically means by, nor categorize all of the biological network motifs into some classes like 'motif mode' in the study by Lee and Tzou [112], where the number of motif modes reaches up to millions. Instead, we assume that biological network motifs are application-dependent, therefore they are flexibly categorized according to the applications. For a specific subgraph being a biological network motif, we need biological measures which will be presented later.

Definition Let $G = (V, E)$ be a graph, and $3 \leq k << |V|$. A **biological network motif** m is a connected subgraph of size k in G, which is biologically significant.

4.3.2 Algorithms

Structural network motifs are either exactly (exhaustively) or approximately determined. As an exhaustive search is infeasible in large networks or for larger size of motifs, approximation algorithms have been used in many applications in practice. In this study, we provide a number of algorithms, which were originally designed to detect biological network motifs but are able to efficiently detect high quality of structural network motifs as well. Some algorithms use structural

information alone or biological information alone, and others combine structural and biological information.

The main idea of the algorithms is to reduce the size of original network so that we can increase the biological network motif detection rates over total number of subgraphs in the original graph. For example, if we remove 22% of edges, then we can reduce the search time by a half as shown in Figure 2.8. We provide two ways of modifying the original network: 1) removing a number of edges and 2) clustering the network into smaller sub-networks. The two methods provide essentially the same components, a list of removed edges and a number of clusters as shown in Figure 4.1. After removing edges, we obtain a number of clusters as by-products; After clustering a network, the edges in between clusters will be the removed before further process.

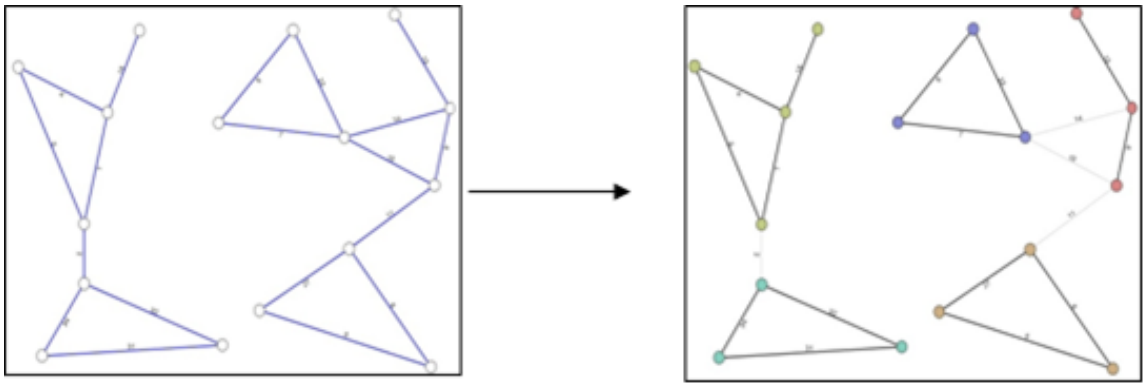


Figure 4.1: After modifying the graph: Original network (left) and the modified network (right) after removing edges or clustering the graph. As shown in the right hand side, a number of clusters and a list of removed edges are provided as a result.

In the following algorithms, $G = (V, E)$ is a target (original) network, $G' = (V, E')$ is a modified network, n is the number of vertices and m is the number of edges in G.

4.3.2.1 Edge-Removing Algorithms

We present two algorithms to remove 'insignificant' edges based on two different aspects. EDGEGO-BNM (EDGEGO for biological network motif) algorithm removes edges based on its related Gene ontology (GO) terms. EDGEBETWEENNESS-BNM (EDGEBETWEENNESS for biological network motif) algorithm removes edges based on its edge betweenness score. Since

EDGEGO-BNM uses GO annotation terms associated with nodes, the algorithm is applicable only to the gene or protein related networks, such as gene regulatory or protein-protein network. EDGEBETWEENNESS-BNM algorithm utilizes existing EDGEBETWEENNESS scoring scheme that has been used for network clustering [230] task, and this is applicable to any biological networks.

EDGEGO-BNM algorithm In EDGEGO-BNM algorithm, we reduce the total number of searches by removing a number of 'biologically insignificant' edges in the original network. Biologically insignificant edges are determined with the GO terms [231] associated with its end points. GO terms provide annotations of gene and gene product attributes across species and databases. GO consists of three independent domains: biological process (BP), molecular function (MF) and cellular component (CC). A BP refers to series of events by multiple molecular functions. Examples include cellular physiological process and pyrimidine metabolic process. An MF is a molecular level of activities, such as catalytic activity or binding. A CC is a component of a cell which is part of larger item. Examples are nucleus, ribosome or proteasome. With the three orthogonal aspects as roots, GO is represented as a directed acyclic graph (GO DAG), a part of which is shown in Figure 4.2. GO DAG describes each GO term as a node and the relationships as an directed edge with hierarchical structure, where children are more specific than the parents. Each term can have multiple parents as well as multiple children and it is traced backward to the root of depth 0. If a gene ge is annotated with a GO term pe, then ge is annotated with all of the ancestor GO terms of pe. Therefore, if two genes are annotated with a GO term with high depth, then the two genes are biologically more related.

We define an **EdgeGO** of an edge e as a set of all GO's associated to both of the end points of e and an **EdgeGOdepth** of e is the maximum depth of the GOs in the EdgeGO, as shown in the following definitions.

Definition Let $G = (V, E)$ be a graph, e be an edge in E and p, q be end points of e. Let g be an GO term in GO DAG.

Let $GO(p)$ and $GO(q)$ are the set of all GO terms associated with p and q respectively. Then,

1. $EdgeGO(e) = GO(p) \bigcap GO(q)$.

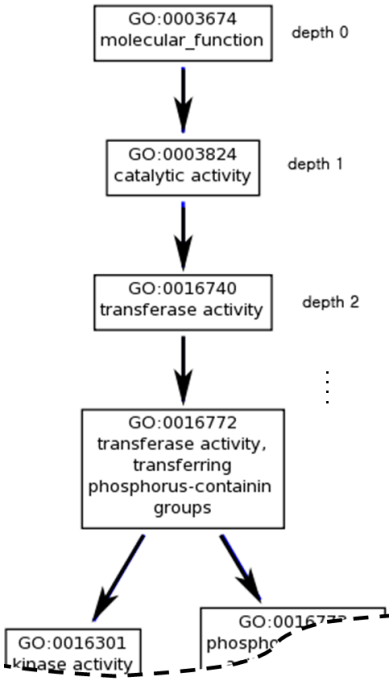


Figure 4.2: An example of GO graph view (GO DAG), where the root node is depth 0. If a GO is depth 0, then it is the most general term, meaning most of genes or proteins are annotated with this GO term. As the depth of the GO increases, the information of GO gets specific.

2. $GOdepth(g) = depth$ of g in DAG graph.

3. $EdgeGOdepth(e) = max\{GOdepth(g) : g \in EdgeGO(e)\}$.

In EDGEGO-BNM algorithm, a threshold GO depth d is given as a parameter and if $EdgeGOdepth(e)$ is less than d, e will be removed. Different d results different number of edges to remove, and we experimentally determine d to get a desired number of subgraphs to search. More edges are removed as d increases, which in turn reduces the search time. This work is motivated by Lee et al. [28] where the authors reveal that different levels of GO terms lead to different motif modes. EDGEGO-BNM is deterministic and the whole process runs linearly with the number of edges in the graph. Algorithm 1 describes the detailed steps of EDGEGO-BNM. Line 11 of Algorithm 1 produces all the k-size subgraphs in the reduced graph G', and any existing exact counting algorithm can be used for this task. In most cases, this algorithm obtains unbalanced

clusters, where a few clusters have most of the vertices and the rest of clusters consist of small number of vertices.

Algorithm 1: EDGEGO-BNM

input : Graph $G = (V, E)$, d :a GO depth threshold
output A number of subgraphs with size k
:
1 $E' \leftarrow E$
2 **for** $\forall e \in E$ **do**
3 p, q be end nodes of e
4 $GO(p)$ = a set of GO terms of p
5 $GO(q)$ = a set of GO terms of q
6 $EdgeGO(e) = GO(p) \bigcap GO(q)$
7 $D \leftarrow EdgeGOdepth(e)$
8 **if** $D < d$ **then**
9 $E' = E' - \{e\}$
10 Let $G' = (V, E')$
11 Enumerate all k-size subgraphs from G'.

EDGEBETWEENNESS-BNM algorithm EDGEBETWEENNESS-BNM algorithm uses topological information to remove some of edges. EDGEBETWEENNESS is initially introduced by Girvan and Newman [230] to produce network clusters using betweenness score of each edge. Network modularization is supported by this method and many protein modules are successfully discovered with EDGEBETWEENNESS [232]. EDGEBETWEENNESS-BNM algorithm, which is specifically designed for network motif discovery, goes through all edges to compute its edge betweenness score, namely, $EBScore$, which is the number of shortest paths in all pairs of vertices that run along with the edge e. Then the edge with maximum $EBScore$ is removed. This process is repeated until we get a desired number of edges to remove. The detail procedure of EDGEBETWEENNESS-BNM is described in Algorithm 2.

Except line 12 of Algorithm 2, EDGEBETWEENNESS-BNM algorithm runs in $O(r|V||E|)$ where r is the number of edges to remove. EDGEBETWEENNESS-BNM algorithm produces relatively balanced network clusters and is also a deterministic algorithm.

Algorithm 2: EDGEBETWEENNESS-BNM

input : Graph $G = (V, E)$, r is the number of edges to remove, k :the motif size.
output a number of subgraphs with size k.
:
1 $RE \leftarrow \emptyset$
2 $E' \leftarrow E$
3 $R \leftarrow 0$
4 **while** $R < r$ **do**
5 for all pairs of vertices in V, obtain the shortest path SP
6 $\forall e \in E$, let $EBscore(e) =$ number of SP's containing e in the path
7 Let ed be the edge with maximum $EBscore$
8 $RE = RE \cup \{ed\}$
9 $E' = E' - \{ed\}$
10 $R = R + 1$
11 Let $G' = (V, E')$
12 Enumerate all k-subgraphs from G'

4.3.2.2 Clustering Algorithms

Another way of reducing a network is to partition the network into smaller sub-networks and remove the edges between clusters. In this work, we present three clustering algorithms: NMF-BNM (Nonnegative matrix factorization for biological network motif), NMFGO-BNM (Nonnegative matrix factorization with GO term for biological network motif) and VOLTAGE-BNM(Voltage clustering for biological network motif) algorithms. Voltage clustering algorithm has been used for network clustering before, but it is the first time to be used for network motif discovery.

NMF-BNM algorithm Nonnegative matrix factorization (NMF) has been used to cluster various data, such as face images, text corpus and gene expression data. Initially used as a dimension reduction technique, NMF is successfully applied to many clustering tasks with additional sparseness constraints [186,213,215]. In this work, we apply NMF for an efficient detection of biological network motifs. Detail process of NMF-BNM is described in Algorithm 3.

In NMF-BNM, a nonnegative matrix $A = (a_{ij})$ of line 4 of Algorithm 3 is topology-based feature data as shown in equation (4.3) and sparseness constraints are added for better clustering. In sparse nonnegative matrix factorization, which appears in line 5 of Algorithm 3, data matrix A

Algorithm 3: NMF(GO)-BNM

input : Graph $G = (V, E)$, c is the number clusters, k :the motif size, (d is GO depth threshold), η and β for sparse NMF.
output a number of subgraphs with size k.
:
1 $RE \leftarrow \emptyset$
2 $E' \leftarrow E$
3 Let $CL_1, \cdots, CL_c = \emptyset$.
4 Construct a data matrix A from G.
5 Run sparse NMF to A and get an $n \times c$ matrix H
6 **for** *all the columns in* H **do**
7 Let $h^j = \{h^j_1, \cdots, h^j_c\}^T$ be jth column vector of H.
8 **if** h^j_i *is largest in* h^j **then**
9 put the vertex v_j to CL_i.
10 **for** $\forall e \in E$ **do**
11 **if** e *lies between clusters of* CL_i **then**
12 RE $= RE \cup \{e\}$
13 $E' = E' - \{e\}$
14 Let $G' = (V, E')$
15 Enumerate all k-subgraphs from G'

is decomposed into two factor matrices W and H using the objective function in Equation (4.4).

$$a_{ij} = \frac{1}{|v_i - v_j|^2}, 1 \leq i, j \leq n \tag{4.3}$$

$$\min_{W,H} \frac{1}{2}\{\|A - WH\|^2_F + \eta\|W\|^2_F + \beta \sum_{j=1}^{m} \|H(:, j)\|^2_1\} \text{subject to } W \geq 0, H \geq 0. \tag{4.4}$$

Here, $\|.\|^2_F$ is the square of the Frobenius norm, $\|.\|^2_1$ of the L_1 norm, and $H(:, j)$ is the j^{th} column of matrix H. Two parameters, η for sparseness and β for balance between sparseness and correctness, should be given. Intuitively, the matrix H gives clustering information as described in lines 6 to 9 of Algorithm 3. The detail computation of sparse NMF is described in the paper by Kim and Park [186]. Except the last step in Algorithm 3, NMF-BNM runs linearly with the size of

A at each iteration, and it converges to a stable point, not necessarily unique, through a number of iterations.

NMFGO-BNM algorithm NMFGO-BNM algorithm differs from NMF-BNM only at the line 4 of Algorithm 3, where the data matrix $A = (a_{ij})$ combines structural information and GO terms of the network as shown in Equation (4.5). In this algorithm, an additional parameter d, which is a GO term depth threshold, is given. First, all the GO terms associated with the network and whose depth is greater than d are listed. Suppose the list of GO terms is $\{g_1, g_2, \cdots, g_p\}$, then each entry a_{ij} in the $(n+p) \times n$ matrix A is defined as in equation (4.5). The rest of process is the same as of the NMF-BNM algorithm.

$$\begin{aligned} a_{ij} &= \frac{1}{|v_i - v_j|^2}, \text{if } 1 \leq i, j \leq n \\ &= 1, \text{if } v_j \text{ is annotated with } g_{i-n} \text{ and } n < i \leq (n+p), 1 \leq j \leq n \\ &= 0 \text{ if } v_j \text{ is not annotated with } g_{i-n} \text{ and } n < i \leq (n+p), 1 \leq j \leq n \end{aligned} \tag{4.5}$$

VOLTAGE-BNM algorithm VOLTAGE clustering algorithm is developed by Wu and Huberman [233] to cluster a network based on voltage drops. The algorithm first generates a number of candidate clusters using Kirchhoff equations [234], which tell that total current of each node should sum up to zero. From the candidate clusters, a seed is selected which appears most frequently in the candidate clusters, and the neighbor vertices of this seed are collected to form a cluster. The process is repeated until we get a desired number of clusters. The number of clusters is later adjusted if the seeds are too close. An exact solution for this algorithm requires O($|V|^3$), but Wu and Huberman [233] provide an approximation solution in $O(|V| + |E|)$. In this work, we utilize VOLTAGE clustering algorithm to design a VOLTAGE-BNM (voltage for biological network motif) algorithm for efficient discovery of biological network motifs as shown in Algorithm 4. We emphasize that VOLTAGE-BNM algorithm is easy and fast, but it is non-deterministic algorithm because the randomly selected seeds lead to quite different results every time it runs. In addition,

we cannot expect to reduce much computation time as not many of the edges can be removed with this algorithm.

Algorithm 4: VOLTAGE-BNM

input : Graph $G = (V, E)$, c is the number clusters, k :the motif size.
output a number of subgraphs with size k.
:
$RE \leftarrow \emptyset$
$E' \leftarrow E$
Let $CL_1, \cdots, CL_c = \emptyset$.
$m \leftarrow 0$.
while *($m \leq c$)* **do**
    `// Generate` c `number of candidate clusters.`
    Pick a vertex pair, *source* and *sink*.
    Compute voltages of each vertex of graph G using *source* and *sink*.
    Group the vertices in two clusters (high/low).
    Store resulting candidate clusters.
    $m = m + 2$
$l \leftarrow 1$
while $l < c$ **do**
    `// generate` $c - 1$ `clusters`
    Pick one cluster seed s most appearing in candidate clusters.
    Obtain co-occurrence vertices to the s, and put them to a cluster CL_l.
    Remove all the co-occurrence vertices and s from candidate clusters.
    $l = l + 1$.
Remaining unassigned vertices belong to the CL_c cluster.
if $\forall e \in E$, *e lies between clusters of* CL_i, **then**
    $RE = RE \cup \{e\}$
    $E' = E' - \{e\}$
Let $G' = (V, E')$
Enumerate all k-subgraphs from G'

Table 4.1 summarizes the algorithms introduced in this chapter. As all of the algorithms have a common step of 'Enumerate all k-subgraphs from G'', the time in this table excludes this last step.

Table 4.1: Various algorithms used for the detection of biological network motifs: All the algorithms introduced in this work are compared based on the type, the time before enumeration by ESU, parameter and its deterministic property. Here d is GO depth threshold, l is the number of GO terms associated to the graph G, c is the number of clusters, r is the number of edges to remove, and η, β for sparse NMF computation.

Algorithm	**Type**	**Time** before ESU	**Parameter**	**Deterministic**
EDGEGO-BNM	Edge-Removing	$O(\|E\|)$	d	Yes
EDGEBETWEENNESS-BNM	Edge-Removing	$O(r\|E\|\|V\|)$	r	Yes
NMFGO-BNM	Clustering	$O(\|E\|(\|V\|+l))$	d, c, η, β	No
NMF-BNM	Clustering	$O(\|E\|\|V\|)$	c, η, β	No
VOLTAGE-BNM	Clustering	$O(\|E\|+\|V\|)$	c	No

4.3.3 Evaluation Methods

Network motif is defined as a frequently and uniquely appearing subgraph in a network and is determined by structural uniqueness testing, measured by P-value (4.1) or Z-score (4.2). The structural uniqueness, however, is an insufficient validation for biological network motifs. Therefore, we design several biological evaluation measures which qualify others rather than topological uniqueness, which are related to protein complexes or functional modules. A protein complex is a group of proteins interacting with each other at the same time and same place in a cell, whereas a functional module is a group of proteins binding to participate in different cellular processes at different times.

The evaluation measures in this work include 'motifs included in complex', 'motifs included in functional module' and 'GO term clustering score'. Currently, these evaluation measures are specifically designed for PPI networks as they require proteins. More comprehensive validation measures should be developed for general biological networks in near future.

Motifs included in complex The first assessment is to check a match with a protein complex. We consider a subgraph g is **included in a complex** if a known protein complex contains all the nodes in g. We define ***motif included in complex*** measure as the precision of the subgraphs included in protein complexes as shown in Equation (4.6). Obviously, the algorithm with higher

value for this measure performs better in this work.

$$\text{Motifs included in complex} = \frac{\text{number of motifs included in a complex}}{\text{number of all discovered subgraphs}} \tag{4.6}$$

Motifs included in functional module Similar to the previous measure, if all components of a subgraph g are included in a known protein functional module, g is **included in a functional module**. Therefore ***motif included in functional module*** is defined as the precision of the subgraphs included in functional modules as in Equation (4.7).

$$\text{Motifs included in functional module} = \frac{\text{number of motifs included in a functional module}}{\text{number of all discovered subgraphs}} \tag{4.7}$$

For the experiments, we can obtain the database for protein complexes and functional modules from MIPS [150] server.

GO term clustering score We define a **P-value of a subgraph** g as the minimum P-value over the union of GO terms of g and lower P-value is preferable. P-value for a GO term is computed using hypergeometric distribution as in equation (4.8), where N is the whole population, M is the population that is annotated by the GO term, n is the subgraph size and x is the number of genes annotated with the GO term in the sample.

$$P - value = \sum_{j=x}^{n} \frac{\binom{M}{j}\binom{N-M}{n-j}}{\binom{N}{n}} \tag{4.8}$$

To determine if a subgraph g with a P-value p is significant, a cutoff value should be predefined. Since P-value decreases as the size of g increases, higher cutoff value is necessary for smaller subgraph. For 4-node and 5-node subgraph, we set the cutoff value as 0.1 and if the P-value of g is lower than the cutoff, g is a *significant* subgraph. A better algorithm will provide more significant subgraphs and lower average p-value of the subgraphs. In other words, the assessment should comprehensively cover both of the P-value and the number of significant subgraphs. To

evaluate the overall performance of an algorithm, we use the ***clustering score*** introduced in the studies of [232, 235] which has measured the quality of a clustering algorithm. For a ***GO term clustering score*** measure, we use subgraphs instead of clusters in the formula of clustering score, as the following.

$$\text{GO term clustering score} = 1 - \frac{\sum_{i=1}^{n_s} min(pi) + (n_i \cdot \text{cutoff})}{(n_s + n_i) \cdot \text{cutoff}}, \tag{4.9}$$

where $min(pi)$ is the P-value of each subgraph, n_s is the number of significant and n_i is the number of insignificant subgraph. A higher GO term clustering score of an algorithm indicates a better algorithm. Since GO term has three independent aspects of BP, MF, CC, we have three types of this measure: BP GO term clustering score; MF GO term clustering score; and CC GO term clustering score.

4.4 Result and Discussion

We test the performance of each algorithm with a couple of PPI of S. cerevisiae (yeast). We download a yeast core data, referred to 'Scere20101010' from DIP database [146] which has 2,130 proteins and 4,434 interactions. We will call this as **DIP Core** network. A network of 988 proteins and 2,455 with high confidence level of interactions, introduced as a high-throughput data in [236] and obtained from the authors of [237], is also used in this experiment. As it was conventionally referred to Y2k, we will also call this **Y2k** network. Since, the increase of network motif size exponentially boosts the computational time, we set the size of subgraphs as four to five for feasible experiments. There are 6 types of non-isomorphic graphs for undirected 4-node subgraphs, and 21 types for undirected 5-node subgraphs. Undirected 4-node subgraph types are labeled using Nauty program [2] as appeared in Figure 4.3.

We first enumerate all subgraphs of size four or five with ESU algorithm [15] and evaluate them with the evaluation measures introduced in this chapter, call it an ESU. Then we run EDGEGO-BNM, EDGEBETWEENNESS-BNM, NMF-BNM, NMFGO-BNM and VOLTAGE-BNM algorithms and measure them with the same evaluation measures. Furthermore, we add the results

by two existing approximation algorithms; RAND-ESU and MFINDER. RAND-ESU searches subgraphs in a tree structure and it skips over some of the branches during its search. MFINDER randomly picks edges until it reaches the desired number of subgraphs. ESU algorithm enumerates all subgraphs and all other algorithms produce roughly 30% of total subgraphs by adjusting parameters. Additionally, we run FANMOD [18], which is a software program implementing ESU, and investigate the topological properties for each type of subgraph in order to observe the relationships between biological network motifs and structural network motifs.

Table 4.2 shows the results for 4-node biological network motifs from DIP core network, with 8 different algorithms measured by its biological meanings; motifs included in complex, motifs included in functional module and GO term clustering scores for BP, MF and CC. The results by ESU, RAND-ESU and MFINDER are also provided as well for comparison purpose. The best result for each measure is marked as bolded in the table. EDGEBETWEENNESS-BNM algorithm provides highest rates for 'motifs included in complex' measure, but EDGEGO-BNM algorithm produces overall the best values compared to others. It is reasonable for the EDGEGO-BNM and NMFGO-BNM algorithms have good scores for GO term clustering score measures as they include GO term information. However, it is interesting to see that EDGEBETWEENNESS-BNM algorithm provides relatively good scores for all of the evaluation measures when this algorithm considers only topological properties of the network. This suggests that the structural property helps infer meaningful biological information as well. We provide the results of 5-node biological network motif search as well in Table 4.3. Similar to the results in Table 4.2, EDGEBETWEENNESS-BNM algorithm is the best for the 'motifs included in complex' term and EDGEGO-BNM is best for the rest of the measures.

To see if the results are consistent with other network, we provide the results with **Y2k** network. The results are shown in Table 4.4 of 4-node subgraph and Table 4.5 of 5-node subgraph. Consistent with DIP core network, EDGEGO-BNM algorithm provides overall good scores except 'motifs included in complex' term and 'MF GO term clustering score'. EDGEBETWEENNESS-BNM algorithm is superior for the 'motifs included in complex' term too. Interesting aspect is that NMFGO-BNM shows good scores as well. Compared to the DIP Core network, the NMFGO-

Table 4.2: Results of 4-node biological network motifs in the *DIP Core*network: We can see that EDGEBETWEENNESS-BNM performs best in 'motif included in complex' measure while EDGEGO-BNM performs best in other measures.

Algorithm	**Motif included in**		**GO Clustering score**		
	Complex	Function	BP	MF	CC
ESU	.13	.205	.64	.51	.61
RAND-ESU	.13	.208	.65	.28	.46
MFINDER	.15	.299	.74	.57	.71
EDGEGO-BNM	.21	**.479**	**.85**	**.70**	**.80**
EDGEBETWEENNESS-BNM	**.28**	.392	.78	.60	.79
NMFGO-BNM	.18	.360	.78	.61	.75
NMF-BNM	.15	.230	.68	.54	.64
VOLTAGE-BNM	.26	.330	.77	.59	.75

Table 4.3: Results of 5-node biological network motifs in the *DIP Core* network: We can see that EDGEBETWEENNESS-BNM performs best in 'motif included in complex' measure while EDGEGO-BNM performs best in other measures.

Algorithm	**Motif included in**		**GO Clustering score**		
	Complex	Function	BP	MF	CC
ESU	.07	.097	.67	.51	.63
RAND-ESU	.07	.096	.66	.52	.62
MFINDER	.09	.167	.75	.56	.72
EDGEGO-BNM	.08	**.240**	**.87**	**.70**	**.79**
EDGEBETWEENNESS-BNM	**.14**	.210	.81	.59	.76
NMFGO-BNM	.08	.169	.71	.59	.60
NMF-BNM	.13	.104	.65	.53	.61
VOLTAGE-BNM	.08	.121	.71	.50	.67

BNM's improved performance in Y2k network can be explained that NMF method performs better with smaller data set. It is also appealing that the random-edge-select algorithm (MFINDER) beats the random-vertex-select algorithm (RAND-ESU). This suggests that edges are more important aspects for explaining its biological meaning.

We also investigate the relationship between structural network motifs and biological network motifs. Table 4.6 is the result generated by FANMOD [18] to observe the statistical properties of each 4-node subgraph type. The first column is the label for each type generated by *Nauty* program [2]. Figure 4.3 shows subgraph shape for each label. Second column indicates the percentage of each type appears in the Y2K network and the next two columns show the average frequencies

Table 4.4: Results of 4-node biological network motifs in the *Y2k* network: We can see that EDGEBETWEENNESS-BNM performs best in 'motif included in complex' measure. NMFGO-BNM performs best on 'MF' and 'CC clustering score' measures. EDGEGO-BNM performs best in the 'motif included in functional module' measure 'BP, CC clustering score' measures. However all the algorithms perform poorly in 'MF clustering score' measure, with less than 30.

Algorithm	**Motif included in**		**GO Clustering score**		
	Complex	Function	BP	MF	CC
ESU	.501	.152	.61	.21	.67
RAND-ESU	.491	.126	.61	.23	.65
MFINDER	.586	.180	.65	.26	.72
EDGEGO-BNM	.603	**.463**	**.94**	.25	**.90**
EDGEBETWEENNESS-BNM	**.904**	.178	.82	.19	.84
NMFGO-BNM	.609	.434	.92	**.27**	**.90**
NMF-BNM	.819	.177	.76	.26	.80
VOLTAGE-BNM	.638	.200	.63	.26	.77

and standard deviation of each type, out of $10,000$ randomized graphs. Last two columns of Z-score and P-value show the structural statistics of each type. As a subgraph of Z-score larger than 2.0 or P-value smaller than 0.01 is a network motif, the three patterns of C$\sim$, C$^\wedge$ and CN are network motifs. We were able to examine that these three patterns were detected as network motifs in the reduced networks as well, as shown in Table 4.7. This is because that each pattern in the reduced network appears with relatively similar frequencies as in the original network. Figure 4.5 shows relative frequencies for each subgraph types, where the horizonal axis lists all six types of non-isomorphic subgraphs and vertical axis indicates its relative frequency. Each line refers to a result of each algorithm, differentiated by colors. All of the algorithms except ESU reduce the total number of searches down to 30%, but the relative frequencies are similar to those by ESU, indicating that our algorithms are applicable to find structural network motifs as well. Same analysis is applied to the DIP Core network as shown in Figure 4.4.

In addition, we provide one example which demonstrates that EDGEGO-BNM algorithm is especially good for discovering motifs included in protein functional modules. This example also shows that structurally non-motifs cannot be ignored as many of the instances are matched with some of protein functional modules. Table 4.8 shows the recall value of 4-node motifs included in a 'rRNA processing' functional module in yeast, based on different subgraph type and algorithms.

Table 4.5: Results of 5-node biological network motifs in the *Y2k* network: We can see that EDGEBETWEENNESS-BNM performs best in 'motif included in complex' measure while EDGEGO-BNM performs best in other measures.

Algorithm	**Motif included in**		**GO Clustering score**		
	Complex	Function	BP	MF	CC
ESU	.281	.083	.69	.17	.76
RAND-ESU	.305	.090	.71	.17	.77
MFINDER	.431	.096	.73	.21	.80
EDGEGO-BNM	.362	**.376**	**.99**	**.24**	**.96**
EDGEBETWEENNESS-BNM	**.814**	.087	.89	.13	.91
NMFGO-BNM	.445	.257	.98	.18	**.96**
NMF-BNM	.643	.073	.80	.18	.83
VOLTAGE-BNM	.665	.089	.82	.19	.85

Table 4.6: Y2k statistical properties, from FANMOD: Each type of 4-node subgraph shows its significance based on its structural uniqueness. The label is generated by *Nauty* program [2] and the corresponding shape is shown in Figure 4.3. In this network, the first three types are detected as network motifs.

Label	Freq(Original)	Mean-Freq (Random)	S-Dev(Random)	Z-score	P-value
C$\sim$	4.66%	2.4634e-006%	4.5133e-071	0	$< 10^{-3}$
C$^\wedge$	8.91%	$o.ooo423\%$	$3.72e-005$	2394.8	$< 10^{-3}$
CN	32.89%	0.021%	0.00139	235.34	$< 10^{-3}$
Cr	0.55%	1.14%	0.00630	-9.48	$> 10^{-2}$
CF	19.58%	41.82%	0.00347	-64.07	$> 10^{-2}$
CR	33.40%	57.02%	0.0029	-80.12	$> 10^{-2}$

We exactly count the numbers of motifs included in 'rRNA processing' with ESU algorithm first. Then all other algorithms are compared with the recall value as computed in Equation (4.10).

$$\text{Recall} = \frac{\text{discovered number of motifs included in a 'rRNA processing' with the algorithm}}{\text{true number of motifs included in a 'rRNA processing'}} \tag{4.10}$$

In Table 4.8, the first column lists different algorithms, and the other columns show the recall in 'rRNA processing' functional module according to each type. The 'rRNA processing' functional module consists of 206 proteins in the yeast. All algorithms except ESU search only 30% of subgraphs out of the total subgraphs searched with ESU algorithm, and EDGEGO-BNM recovers over

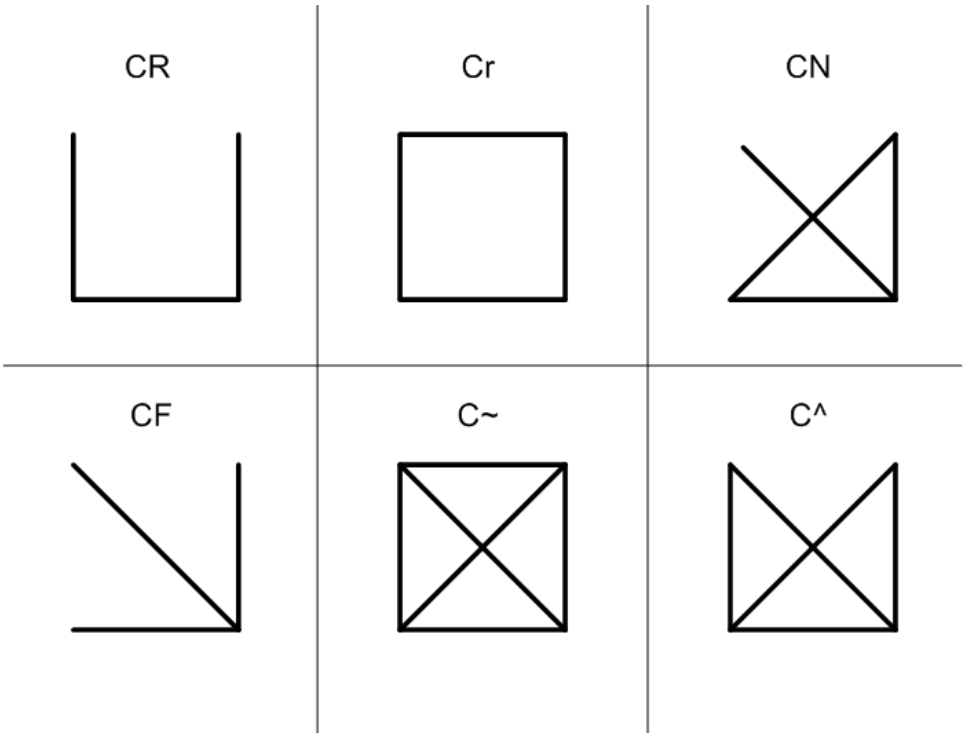


Figure 4.3: Shapes and labels for 4-node subgraphs in an undirected network: There are six types for 4-node subgraphs in an undirected network. Each type is labeled by *Nauty* program.

90% of subgraphs included in 'rRNA processing'. We can also see that although the Cr, CF, CR types are structurally insignificant, about 50% of subgraphs included into the 'rRNA processing' are these non-motifs. This example shows that even non-motifs also have biological meanings, therefore the structural network motif defined by its structural uniqueness is insufficient to explain biological meanings.

4.5 Summary and Future Work

In this work, we provide new approaches to finding network motifs in biological networks. Structural network motifs are defined as frequently and uniquely repeated small connected subgraphs in a network. However, motivated by several issues with a number of network motif applications, we suggest to search biologically meaningful network motifs. Hence, we define a **biological network motif** as a biologically meaningful k-node subgraph, develop a number of algorithms for efficient detection of biological network motifs and introduce new evaluation measures to validate

Table 4.7: Y2k reduced network by EDGEGO-BNM statistical properties, from FANMOD: Each type of 4-node subgraph shows its significance based on its structural uniqueness. The label is generated by *Nauty* program [2] and the corresponding shape is shown in Figure 4.3. In this network, the first three types are detected as network motifs.

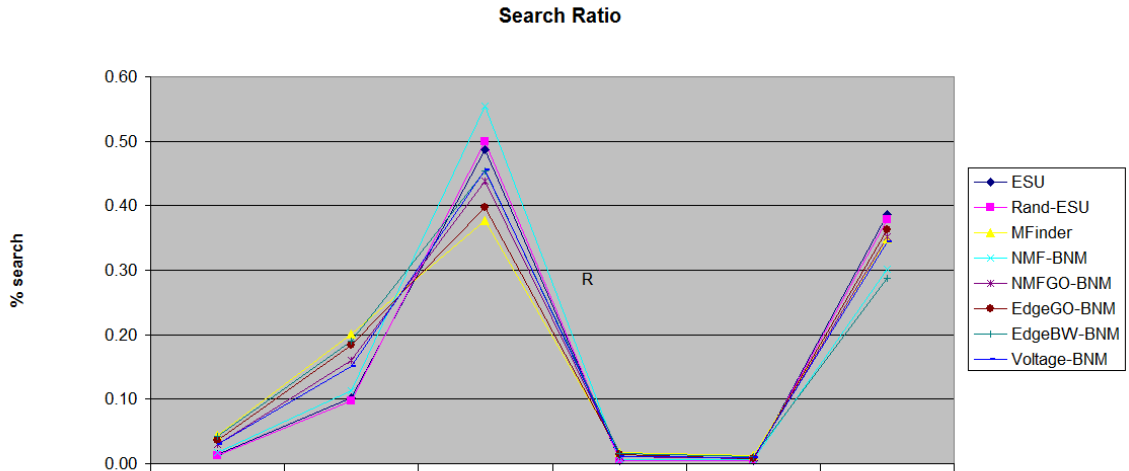
Label	Freq(Original)	Mean-Freq(Random)	S-Dev(Random)	Z-score	P-value
C$\sim$	7.09%	$4.54E-08$	$9.18E-07$	77220	$< 10^{-3}$
C$^\wedge$	11.54%	0.00%	5.18E-05	2226.7	$< 10^{-3}$
CN	35.24%	0.14%	0.0016036	218.86	$< 10^{-3}$
Cr	0.75%	1.34%	0.00079905	-7.3172	$> 10^{-2}$
CF	16.41%	40.70%	0.003686	-65.909	$> 10^{-2}$
CR	28.98%	57.82%	0.0029743	-96.969	$> 10^{-2}$

Table 4.8: The rates of motifs included in a 'rRNA processing' functional module in the yeast (Y2k network), computed using Equation (5.18): Except ESU, all algorithms search 30% of subgraphs in the original network. However, EDGEGO-BNM recovers over 90% of motifs included in functional module. We note that the non-motif types of Cr, CF and CR have a number of instances for this functional match, indicating structural uniqueness is insufficient to discover its biological significance.

Algorithm	**C$\sim$**	**C$^\wedge$**	**CN**	**Cr**	**CF**	**CR**
ESU	1.0	1.0	1.0	1.0	1.0	1.0
(Counts)	(2,509)	(5,152)	(17,457)	(434)	(8,095)	(15,953)
RAND-ESU	.30	.32	.34	.36	.34	.34
MFINDER	.78	.54	.31	.38	.16	.13
EDGEGO-BNM	**.97**	**.97**	**.98**	**1.0**	**.99**	**.97**
EDGEBETWEENNESS-BNM	.67	.64	.32	.57	.22	.16
NMFGO-BNM	.87	.88	.78	.89	.70	.73
NMF-BNM	.69	.39	.23	.22	.12	.90
VOLTAGE-BNM	.53	.38	.39	.39	.32	.31

biological qualities of motifs. The algorithms reduce the number of subgraph search and increase the detection rates of biological network motifs at the same time. The algorithms are categorized into two classes: Edge-removing algorithms and network clustering algorithms. EDGEGO-BNM and EDGEBETWEENNESS-BNM are algorithms which remove a number of edges based on GO terms and edge betweenness scores, respectively. NMF-BNM, NMFGO-BNM and VOLTAGE-BNM algorithms partition the network based on its topological property or GO term relevance.

We also introduce a number of evaluation measures which validate biological significance of each subgraph: 'motifs included in complex', 'motifs included in functional module' and 'GO

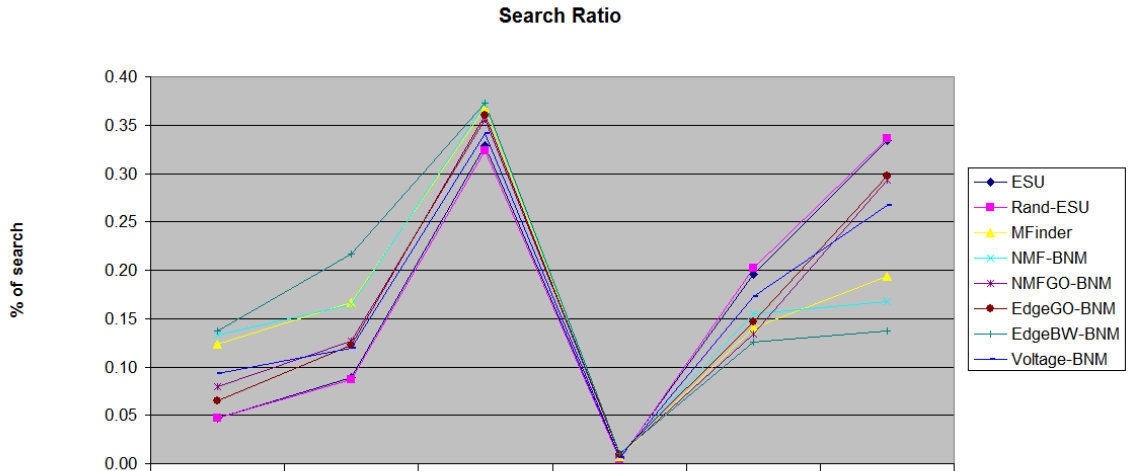


	<C^>	<CN>	<CF>	<C~>	<Cr>	<CR>
ESU	0.01	0.10	0.49	0.00	0.00	0.39
Rand-ESU	0.01	0.10	0.50	0.00	0.00	0.38
MFinder	0.04	0.20	0.38	0.02	0.01	0.35
NMF-BNM	0.02	0.11	0.55	0.01	0.01	0.30
NMFGO-BNM	0.03	0.16	0.44	0.01	0.01	0.35
EdgeGO-BNM	0.04	0.18	0.40	0.01	0.01	0.36
EdgeBW-BNM	0.04	0.19	0.45	0.02	0.01	0.29
Voltage-BNM	0.03	0.15	0.46	0.01	0.01	0.34

Figure 4.4: DIP Core network: Search ratios based on the subgraph type: The ratio of frequency of each type is relatively preserved and it indicates that our algorithms can be used for the structural network motif discovery as well. Relative frequencies of each algorithm is plotted with different colors of line. The horizontal axis indicates each subgraph type for 4-node subgraphs. The vertical axis shows the relative frequency of each type. The values are shown in the table below the figure.

term clustering score.' Biological meanings can be assigned to biological network motifs based on these evaluation measures. We ran the algorithms on two PPI network of *S.cerevisiae*, and compared them with our new measures. An existing exhaustive search and other two existing approximation algorithms are also provided to be competed with our algorithms. EDGEGO-BNM shows overall good results but EDGEBETWEENNESS-BNM is the best in terms of the 'motifs included in complex' measure. We were also able to show that all these algorithms can be alternatives of existing network motif algorithms.

This work has three contributions to the study of network motifs: 1) We question biological meanings of network motifs which have not been focused by existing detection algorithms. New motif search algorithms and evaluation measures are developed based on these questions. 2) We



	<C~>	<C^>	<CN>	<Cr>	<CF>	<CR>
ESU	0.05	0.09	0.33	0.01	0.20	0.33
Rand-ESU	0.05	0.09	0.32	0.00	0.20	0.34
MFinder	0.12	0.17	0.37	0.01	0.14	0.19
NMF-BNM	0.13	0.16	0.37	0.01	0.16	0.17
NMFGO-BNM	0.08	0.13	0.36	0.01	0.13	0.29
EdgeGO-BNM	0.06	0.12	0.36	0.01	0.15	0.30
EdgeBW-BNM	0.14	0.22	0.37	0.01	0.13	0.14
Voltage-BNM	0.09	0.12	0.34	0.01	0.17	0.27

Figure 4.5: Y2k network: Search ratios based on the subgraph type: The ratio of frequency of each type is relatively preserved and it indicates that our algorithms can be used for the structural network motif discovery as well. The description of the plots and the table is same as in Figure 4.4.

design several algorithms combining the topological and biological information in a network. The algorithms further enrich existing algorithms in biological contexts. 3) We develop a number of evaluation measures which qualify biological importance of network motifs. As we know of, this is the first time to suggest systematical evaluation measures for network motifs.

The works in this chapter can be studied further. Currently, the parameters of various algorithms in this work are adjusted only to obtain a desired number of subgraphs. In near future, various impacts of the parameters on the results should be investigated. Besides the parameters, the balance between topological and biological information will be an important factor for a better algorithm. On the other hand, current evaluation measures are limited to PPI networks. Comprehensive evaluation measures should be designed to apply various types of biological networks.

Meanwhile, the work should be extended to weighted or direct networks for more comprehensive analysis of biological network motifs.

Chapter 5

ESSENTIAL PROTEIN DISCOVERY IN A PPI NETWORK USING NETWORK MOTIF AND GENE ONTOLOGY

5.1 Background

Essential proteins are the proteins that are crucial for development to a fertile adult and the functions of them are vital to a cellular life [238–240]. Removal of any one of essential proteins leads to a fatal defect on an organism [35]. Therefore, identification of essential proteins helps understand cellular life of an organism as well as use for actual usages including drug design [241]. Essential proteins have been identified through experimental procedures such as genetic screens [242], single gene knockouts [243], RNA interference [244] and conditional knockouts [245]. The experimental techniques have identified enough essential and non-essential proteins for databases such as DEG [246] and SGD [247]. Therefore, thanks to these databases and because of many limitations of experimental techniques, computational approaches have been recently suggested [35, 248–250].

Computational approaches vary from sequencing analysis [251, 252] to network analysis, which involve machine learning algorithms [35], graph theory and centrality measurements [144, 253, 254]. In most cases, the methods using centrality measurements determine essential proteins in a protein-protein interaction (PPI) network, which is an undirected graph where proteins are nodes and their binary interactions are edges. By utilizing topological features in a network, many centrality algorithms have been developed and used for identifying essential proteins, including degree centrality (DC) [144, 255, 256], betweenness centrality (BC) [257, 258], closeness centrality (CC) [259], subgraph centrality (SC) [260] and eigenvector centrality (EC) [261]. It has been shown that these centrality algorithms perform greatly better than random selection [262], and they have been compared in the studies by Wang et al. [254] and Li et al. [263].

5.2 Problem Statement

Existing centrality algorithms have two problems. First, most centrality algorithms are very sensitive to false links in networks, where false links are unavoidable as current networks are still growing and incomplete. Secondly, existing centrality algorithms focus only on the structural features of networks, meaning that biological properties are not considered in the process.

Therefore, in this chapter, we want to develop a robust centrality algorithm which involves biological information. The algorithm uses network motifs in a GO-(Gene ontology)-pruned network with EDGEGO algorithm, named ***MCGO***. Network motifs, motivated by protein sequence motifs, are over represented small connected subgraph patterns in a network. Ever since they were introduced as functional building blocks in a transcriptional regulatory network [16], network motifs have been used in many biological applications including identification of specific genes [107], prediction of protein-protein interactions [26] and examination for the relationship with evolutionary conservation [27]. In this work, we utilize network motifs for a new centrality measure and name it ***Motif Centrality (MC)***. MC is more robust than other centrality measures as network motifs are rarely affected by false links in networks due to their nature of statistical uniqueness. Additionally, we apply EDGEGO algorithm to the original network to involve biological information in MC. EDGEGO algorithm removes some of 'biologically insignificant' edges from a PPI network based on Gene Ontology (GO) annotation terms [231] and produces a *GO-pruned* network. Then we define a new centrality algorithm, ***MCGO***, which is MC in the GO-pruned network resulted by EDGEGO. In fact, GO terms were previously used as some features in determining essential genes using machine learning techniques by Acencio and Lemke [35]. But in MCGO method, GO terms are indirectly but efficiently incorporated in the edge-pruning process of the network.

We investigate the performance of MCGO by testing it with a *Saccharomyces cerevisiae* PPI network as essential and non-essential proteins are well classified in this organism. The PPI network is downloaded from DIP server [146] and each set of essential proteins and non-essential proteins are collected from MIPS [150], SGD [247], SGDP [264] and DEG [246] databases. Previously, Wang et al. [254] and Li et al. [263] have proposed new centrality algorithms, SoECC

(sum of edge clustering coefficient) and LAC (local average connectivity) respectively, compared their method with a number of other centrality algorithms and showed that their new method performs better than others based on various validation measures. Similarly, we also compare MCGO with other existing centrality algorithms of DC, BC, CC, SC, EC, SoECC and LAC, and prove that MCGO is more effective than others. The work in this chapter is an extension of Kim et al. [34] where MCGO is compared with DC and SoECC only. All of the algorithms are evaluated based on the following three evaluation methods; 'Top-ranked (TR) proportion,' 'Statistical measures including sensitivity (SN), specificity (SP), F-measure (F), positive predictive value (PPV), negative predictive value (NPV) and accuracy (ACC),' and 'precision-recall (PR) curve'. Experimental results demonstrate that MCGO performs the best in all the evaluation measures. Additionally, we observe that if other centrality algorithms are performed in a GO-pruned network by EDGEGO, then their performances improve in a great amount as well.

Overall, the work has two contributions in the task of discovery of essential proteins: 1) Network motifs are used in the application of the detection of essential proteins as the first time. We showed that network motifs are robust and effective tools for this task. 2) We incorporate biological information, which is GO annotation terms, into the process of centrality ranking. It is also the first attempt to rank the vertices based not only on the network topology, but also on the biological information. Interestingly, the incorporation of GO terms into other centrality algorithms affects the performance in a positive way.

5.3 Methods

In this section, we first review existing centrality algorithms then introduce a new centrality algorithm. The new algorithm is compared with other algorithms based on three measures, which will be also described in this section.

5.3.1 Algorithms

Existing centrality algorithms are instable in incomplete networks and they are derived only by structural properties. Therefore, we develop a robust and biologically meaningful centrality

algorithm using network motifs and gene ontology terms. To see how the algorithms are different and why we need a new algorithm, we review existing centrality algorithms, then introduce MC and MCGO algorithms.

5.3.1.1 Centrality Algorithms

Centrality algorithms are useful to determine more influential individuals from a social group which is represented as a social network [13]. Many researchers applied centrality measures to analyze biological networks such as prediction of essential proteins in PPI networks [254,265,266] and detection of global gene regulator in gene regulation networks [267]. However the term of '*centrality*' is ambiguous as the notion depends on the context. For example, as shown in Figure 5.1, a vertex would be central if the network is separated into two or more components with the removal of the vertex. On other hand, a vertex is central if the network is scattered when the vertex is removed. Therefore, various centrality algorithms are developed with different purposes and interpretations.

Wang et al. [254] introduced a new centrality algorithm, named SoECC (sum of edge clustering coefficient centrality), for identifying essential proteins in PPI networks, and compared the performance with those of other existing centrality algorithms. The study showed that while any one of the other algorithms is not dominantly good, SoECC outweighs all other centrality methods based on several validations. Similarly, Li et al. [263] introduced LAC, defined as a local average connectivity, and showed its superior performance compared with other methods as well. In this chapter, we introduce MC (Motif Centrality) and MCGO (Motif Centrality in GO-pruned network) and compare the performance with those of DC, BC, CC, SC, EC, SoECC and LAC.

For the sake of notation, a PPI network is regarded as an undirected graph $G = (V, E)$ where V is the set of vertices (proteins) and E is the set of edges (interactions). The number of vertices in G is N and A is defined as the adjacency matrix of the network G. Each node $u \in V$ is ranked differently with different centrality algorithms as the followings;

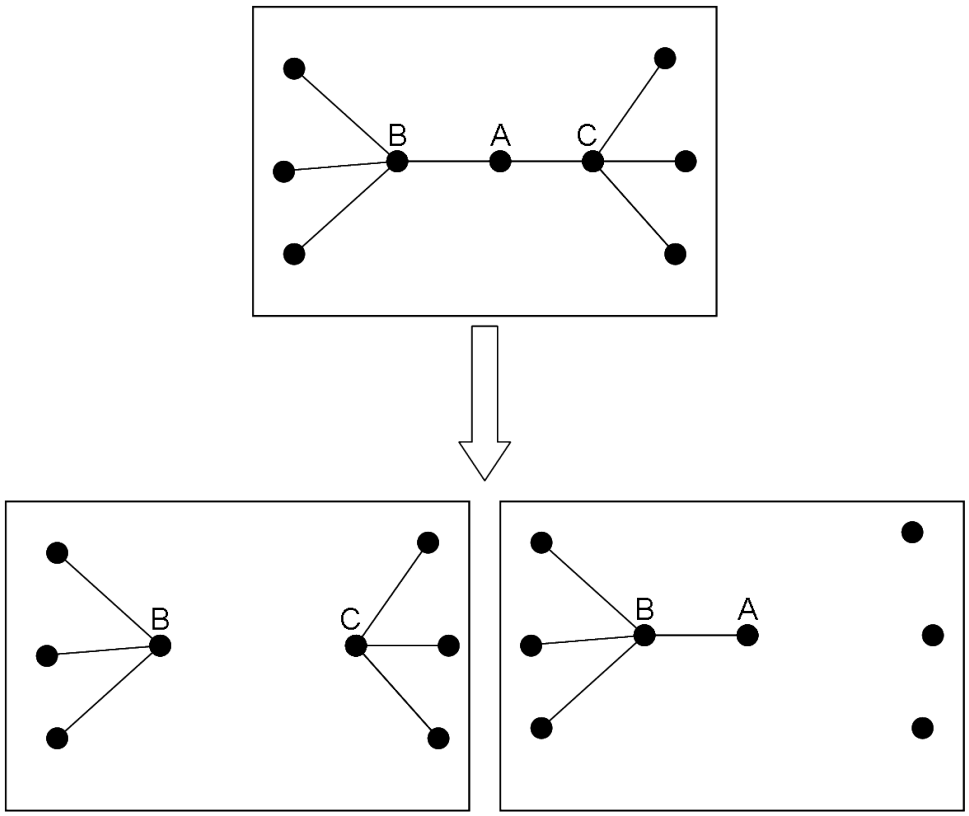


Figure 5.1: The top graph is an original network. If we remove A, then the graph is separated into two subgraphs as shown in the bottom-left side. However, if we remove B or C, the graph is nearly scattered as appeared in the bottom-right side. Therefore, a central node is not deterministic. The graph is captured from [13].

1. Degree Centrality (DC) ranks each node as its degree.

$$DC(u) = d_u \tag{5.1}$$

   where d_u is the degree of u in G.

2. Betweenness Centrality (BC) determines the score of a node u as average fraction of the shortest paths passing through u.

$$BC(u) = \sum_s \sum_t \frac{\rho(s,u,t)}{\rho(s,t)}, s \neq t \neq u \tag{5.2}$$

where $\rho(s,t)$ is the total number of the shortest paths of s and t and $\rho(s,u,t)$ is the number of the shortest paths of s and t which includes u in the path.

3. Closeness Centrality (CC) of a node u is the inverse proportion of the average of graph-theoretic distances from u to all other nodes in G.

$$CC(u) = \frac{N-1}{\sum_v dist(u,v)} \tag{5.3}$$

where $dist(u,v)$ is the distance between u and v, which is the number of links in the shortest path of u, v.

4. Subgraph Centrality (SC) of a node u is defined as the number of subgraphs in G where u participates. The smaller the subgraph is, the more weights are given.

$$SC(u) = \sum_{l=0}^{\infty} \frac{\mu_l(u)}{l!} = \sum_{v=1}^{N} [\alpha_v(u)]^2 e^{\lambda_v} \tag{5.4}$$

where $\mu_l(u)$ is the number of closed loops of length l at u. $\alpha_i, (1 \leq i \leq N)$ is the orthonormal basis of R^N composed by eigenvectors of A, associated to the eigenvalues of $\lambda_j, (1 \leq j \leq N)$. Here, $\alpha_v(u)$ is the u^{th} component of α_v.

5. Eigenvector Centrality (EC) of a node u is the u^{th} component of the principal eigenvector of A.

$$EC(u) = \alpha_{max}(u) \tag{5.5}$$

where α_{max} is the eigenvector corresponding to the largest eigenvalue of A, and $\alpha_{max}(u)$ is the u^{th} component of α_{max}.

6. Sum of edge clustering coefficient (SoECC) is the sum of all neighborhood edge clustering coefficients.

$$SoECC(u) = \sum_{v \in N_u} ECC(u,v) \tag{5.6}$$

$$= \sum_{v \in N_u} \frac{z_{u,v}}{min(d_u - 1, d_v - 1)} \tag{5.7}$$

where N_u is the set of all neighbors of u, $Z_{u,v}$ is the number of triangles that include the edge (u,v), d_u and d_v are degree of u and v in G, respectively.

7. Local Average Connectivity (LAC) of a node u is defined as the average local connectivity of its neighbors.

$$LAC(u) = \frac{\sum_{w \in N_u} deg^{C_u}(w)}{|N_u|} \tag{5.8}$$

where N_u is the set of neighbors of node u, and C_u is the subgraph induced by N_u. $deg^{C_u}(w)$ is the degree of w in the graph C_u.

5.3.1.2 Motif Centrality (MC) and MCGO

Network motifs are defined as frequent and unique subgraph patterns in a network and they are used in many biological applications. Similar to a protein sequence motif, network motif is defined as an overly repeated pattern, but the detection process requires much costly computation as it involves NP-hard isomorphic testing and repeated processes for uniqueness determination. Definition 4.3.1 defines network motif m formally.

We define a **Motif Centrality** of a node u, MC(u), as the number of motifs where u is contained, divided by a weight w_k, as defined in Equation (5.9). **MCGO(u)** in Equation (5.10) is MC(u) in a reduced graph G' which is the result of EDGEGO(G). EDGEGO algorithm is described in the next section.

$$MC(u) = \sum_{i=1}^{n} \frac{m_i(u)}{w_k}, u \in G \tag{5.9}$$

$$MCGO(u) = MC(u), u \in G' \tag{5.10}$$

In the above equations, n is the number of all the network motifs in G, k is the motif size and w_k is $|V|^k$, and $m_i(u) = 1$ if the node u is a member of the motif m_i, otherwise $m_i(u) = 0$. The size of motif k is currently set to 3 or 4 for practical usage.

We should note that MC is closely related to DC and SC as well. In most cases, if a node has a high degree, then the node has a higher chance of being involved in more network motifs. However, MC is more complicated than DC because MC is affected not only by directed neighbors but also by neighbors with several hops. And MC is more robust than SC as network motifs are frequent and unique subgraphs, while SC involves all the subgraphs regardlessly.

5.3.1.3 EDGEGO algorithm

Most of centrality algorithms are based on the structure of the network only, such as degree, distance or edge clustering coefficient. In this work, we introduce an algorithm, EDGEGO, which removes a number of 'biologically insignificant' edges from the network, as a method to incorporate biological information. EDGEGO algorithm is similar to EDGEGO-BNM in Algorithm 1 which was used to detect biological network motifs in Chapter 4. The only difference is that EDGEGO in Algorithm 5 stops when it removes a number of edges with GO terms and returns a reduced network , while EDGEGO-BNM processes further to detect network motifs.

In EDGEGO algorithm, Gene ontology (GO) [231] terms for the proteins in a network determine biologically insignificant edges to be removed. We specifically utilize GO in a PPI network, as GO terms provide annotations of gene and gene product attributes across species and databases. GO consists of three independent domains: biological process (BP), molecular function (MF) and cellular component (CC). A BP refers to series of events by multiple molecular functions, such as, cellular physiological process and pyrimidine metabolic process. An MF is a molecular level of activities, including catalytic activity or binding. A CC is a component of a cell which is part of larger item. Nucleus, ribosome or proteasome are the examples. GO is represented as a directed

acyclic graph (DAG) as shown in Figure 4.2, so each GO term has its informative depth in GO DAG. All the proteins in a data network are annotated with multiple GO terms as if a gene ge is annotated with a GO pe, it means ge is annotated with all the ancestor GO terms of pe.

We define an **EdgeGO** of an edge e as a set of all GO's associated to both of the end points of e and an **EdgeGOdepth** of e is the maximum depth of the GOs in the EdgeGO, as shown in the Definition 4.3.2.1.

Algorithm 5: EDGEGO

input : Graph $G = (V, E)$, d :a GO depth threshold
output Reduced graph $G' = (V, E')$.
:
$E' \leftarrow E$
for $\forall e \in E$ **do**
    p, q be end nodes of e
    $GO(p)$ = a set of GO terms of p
    $GO(q)$ = a set of GO terms of q
    $EdgeGO(e) = GO(p) \bigcap GO(q)$
    $D \leftarrow EdgeGOdepth(e)$
    if $D < d$ **then**
        $E' = E' - \{e\}$
Output $G' = (V, E')$

Algorithm 5 describes the detailed steps of EDGEGO, where a threshold d should be given as a parameter and if $EdgeGOdepth(e)$ is less than d, e will be removed. Different d results different number of edges to remove, and we experimentally determine d for the best results. This work is motivated by Lee et al. [28] which reveals that different levels of GO terms lead to different motif modes. EDGEGO is deterministic and the whole process runs linearly with the number of edges in the graph.

5.3.2 Evaluation Methods

MCGO is compared with other centrality algorithms in various validation measures and we provide 'TR proportion,' 'Statistical measures,' and 'Precision and Recall curve' methods as the followings.

5.3.2.1 TR proportion

We rank proteins in a decreasing order by each centrality algorithm and determine top ranked proteins as predicted essential proteins or a candidate set. The size of candidate set is varied with top 5%, top 10%, top 15% and top 20% of all the proteins in the data. At each candidate set, we determine the rate of actual essential proteins in the candidate set, which we name as 'Top-Ranked (TR) proportion.' The algorithm with the highest TR-proportion values will be considered as the best.

5.3.2.2 Statistical measures

Detection of essential proteins is a binary decision problem as each protein is either labeled as essential (positive) or non-essential (negative) if we ignore unknown proteins. Therefore, each centrality algorithm is a classifier and each decision is represented as a confusion matrix or contingency table as shown in Table 5.1. The confusion matrix has four terms and they are interpreted as the followings.

- TP (true positive): Essential proteins correctly predicted as essential.
- FP (false positive) : Non-essential proteins incorrectly predicted as essential.
- TN (true negative): Non-essential proteins correctly predicted as non-essential.
- FN (false negative): Essential proteins incorrectly predicted as non-essential.

For statistical assessments, we compare each algorithm based on sensitivity (SN), specificity (SP), positive predictive value (PPV), negative predictive value (NPV), F-measure (F) and accuracy (ACC) measures, as the followings:

1. Sensitivity (SN) refers to the number of correctly predicted essential proteins over the total number of essential proteins.

$$SN = \frac{TP}{TP + FN} \tag{5.11}$$

2. Specificity (SP) is the number of correctly predicted non-essential proteins over the total number of non-essential proteins.

$$SP = \frac{TN}{TN + FP} \tag{5.12}$$

3. Positive Predictive Value (PPV)is the number of correctly classified essential proteins over the total number of predicted essential proteins.

$$PPV = \frac{TP}{TP + FP} \tag{5.13}$$

4. Negative Predictive Value (NPV) means the proportion of correctly predicted non-essential proteins over the predicted non-essential proteins.

$$NPV = \frac{TN}{TN + FN} \tag{5.14}$$

5. F-measure (F) refers the harmonic mean of SN and PPV.

$$F = \frac{2 \cdot SN \cdot PPV}{SN + PPV} \tag{5.15}$$

6. Accuracy(ACC) is the number of all correctly predicted proteins, positively or negatively, over the total number of known proteins.

$$ACC = \frac{TP + TN}{P + N} \tag{5.16}$$

where P and N is the number of essential proteins and non-essential proteins, respectively.

5.3.2.3 Precision-Recall Curve

For comprehensive comparison, we propose to use a precision-recall curve which is a more inclusive evaluation measure. Simply presenting accuracy results when performing an empirical

validation is insufficient as accurate rates depend on each candidate size. Provost et al. [268] also argued that the comparison based on accuracy results can be misleading. In our study, some algorithms provide higher portion of true positive when the candidate set is small; Others perform better when the candidate set is large. Therefore, in order to compare the overall performance of each algorithm, Precision-Recall (PR) curve is more appropriate and the area under curve value is a representative score. PR curves are often used in information retrieval study [269] as an alternative to Receiver Operator Characteristic (ROC) curve for the problems in a large skewed data sets [270–272]. PR curve is more proper than ROC in this task as our data set is largely skewed with undistributed number of essential and non-essential proteins.

In this study, the precision and recall is defined as in the Equation (5.17) and Equation (5.18).

$$Precision = \frac{\text{\# of (essential proteins} \bigcap \text{candidates)}}{\text{\# of candidates}} \tag{5.17}$$

$$Recall = \frac{\text{\# of (essential proteins} \bigcap \text{candidates)}}{\text{\# of essential proteins}} \tag{5.18}$$

When we predict essential proteins, if we vary the size of candidate linearly then we can see a trend in a PR curve. In PR space, x-axis is *Recall* and y-axis is *Precision* and each point in the curve represents a pair of recall and precision value at a given candidate size. The goal of a PR curve is to be in the upper-right hand corner. Each algorithm is plotted as one PR curve and the one with the most upper-right hand is considered as the best algorithm. Most cases, however, visual comparison is unclear. Therefore the AUC (area-under-curve) value is typically used as a measure.

5.4 Results and Discussion

5.4.1 Experimental data

We test the performances of MC and MCGO, and examine the effect of EDGEGO for the discovery of essential proteins with a *Saccharomyces cerevisiae* (Baker's yeast) PPI data. The

data, named, *Scere20101010*, is downloaded from DIP server [146] which contains 5,197 proteins, 25,229 interactions. We also collect 1,316 essential proteins and 4,383 non-essential proteins from DEG [246], MIPS [150], SGD [247] and SGDP [264] databases. Out of 5,197 proteins in our data set, 1,202 proteins are essential, 3,610 are non-essential and 386 are unknown. For comparison, other methods including DC, BC, CC, SC, EC, SoECC and LAC are also processed with the same data and the results are given.

5.4.2 Evaluation by three evaluation measures

MC and MCGO are compared with other existing seven algorithms based on the 'TR proportion,' 'Statistical measures,' and 'PR-curve' evaluation methods introduced in the section 5.3.2. The results based on 'TR proportion' are shown in Figure 5.2 where the Y-axis is the rate of correctly predicted essential proteins and X-axis is the candidate size as percentages over the whole date set. The bars are grouped according to each candidate set and each bar indicates the correctly predicted rates of DC, BC, CC, SC, EC, SoECC, LAC, MC and MCGO from left to right. We can observe that MCGO performs best in this evaluation measure.

All the algorithms are nextly compared based on six statistical measurements as shown in Figure 5.3. The bars are grouped by each validation measure, SN, SP, PPV, NPV, F and ACC, in that order. None of the algorithms except MCGO holds the dominant position. However, MCGO beats all the other algorithms in all the statistical measures, which is consistent with the result of Figure 5.2.

Precision-Recall (PR) curves are also provided in Figure 5.4 to see overall performance, and it demonstrates that MCGO performs the best as it is located in the most upper-right hand corner. For clear comparison results, the area-under-curve (AUC) values for Precision-Recall curves are presented in Table 5.2. With the AUC value, we can rank the algorithms in the order of MCGO, SoECC, LAC, MC, DC, SC=EC, BC and CC.

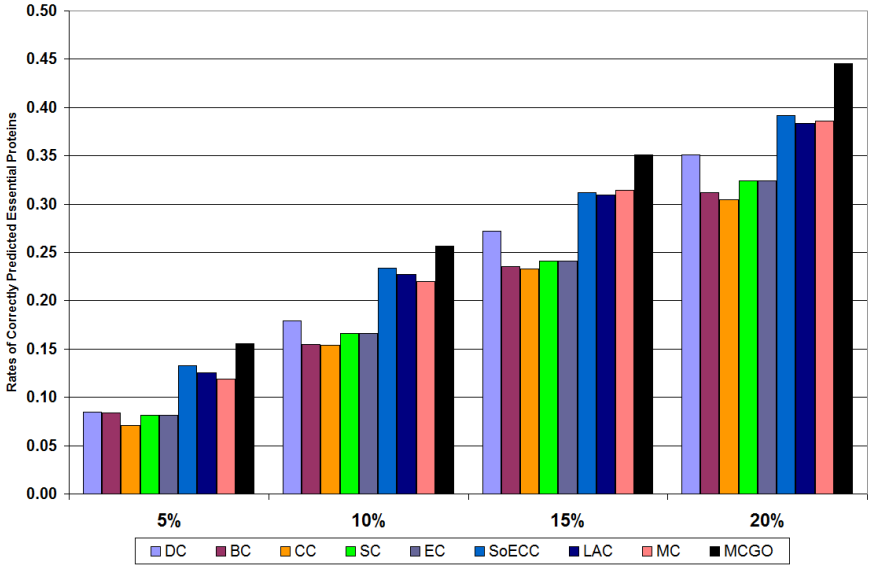


Figure 5.2: TR proportion: Each bar indicates the performance result of DC, BC, CC, SC, EC, SoECC, LAC, MC and MCGO from the left to right.

5.4.3 Effects of EDGEGO

We observed that involvement of GO improves the performance of detecting essential proteins in a network, as shown with the results by MCGO and MC in the previous section. Therefore, we examined the effects of EDGEGO to other algorithms as well. We name the EDGEGO involved algorithms as -*GO* algorithms, such as, DCGO, BCGO, CCGO, SCGO, ECGO, SoECCGO and LACGO. As a result, all -*GO* algorithms perform better than original algorithms in all three evaluation methods, which is shown as the increased AUC values of Table 5.2. We also provide the results in graphs from Figure 5.5 to Figure 5.10. Figure 5.5 and Figure 5.8 shows that all -*GO* algorithms improve the TR-proportion rates against their original algorithms. Likewise, the -*GO* algorithms perform better than the original in statistical measures as appeared in Figure 5.6, Figure 5.9 and in PR curves as shown in Figure 5.7 and Figure 5.10. However, MCGO is still superior to all other EDGEGO involved algorithms as shown in Figure 5.11, Figure 5.12, Figure 5.13 and the AUC value in Table 5.2.

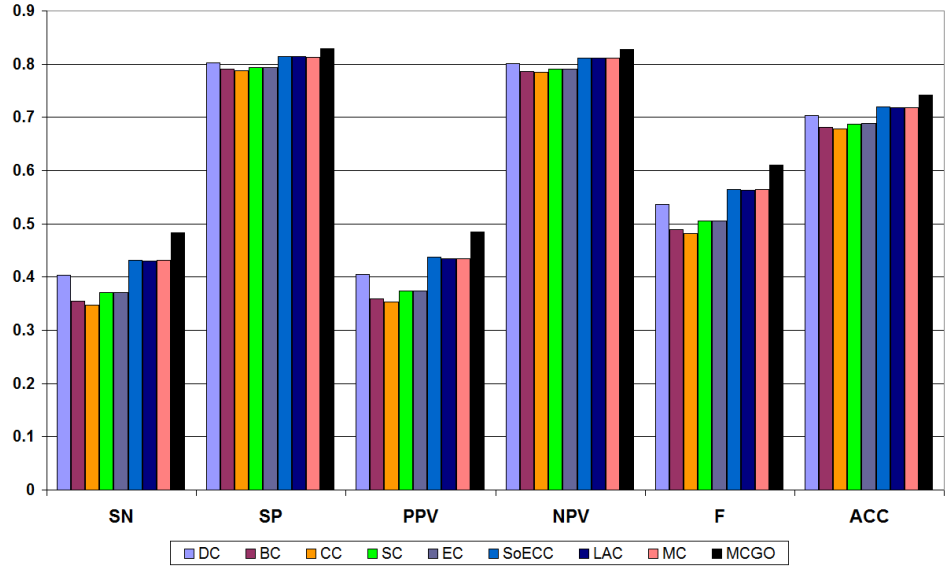


Figure 5.3: Statistical measures including SN, SP, PPV, NPV, F and ACC: Each bar indicates the performance result of DC, BC, CC, SC, EC, SoECC, LAC, MC and MCGO from the left to right.

5.4.4 Analysis of MC and MCGO

Figure 5.14 explains one example of the process of MC in detecting essential proteins. ALG1 is an essential gene and produces one of proteins in our data. As we can see in Figure 5.14(a), the degree of ALG1 is small, therefore DC classifies it as a non-essential protein, and SoECC also likely to predict it as a non-essential as SoECC depends on the number of neighbor vertices and edges. On the other hand, MC considers not only the number of neighbors but also its structural uniqueness in a network. In the network, there are two types of 3-node subgraph, and a type of triangle shape is determined as a network motif. When we expand the node WBP1 which is a neighbor of ALG1, some of ALG1's other neighbors such as ELO3, ELO2, PHO88, YET1 are connected to it, making a new structure as appeared in Figure 5.14(b). In this way, whenever the triangle structure is formed, the node's MC rank increases, therefore, even with small degree, MC most likely classifies it as an essential protein. If the network motif size is large, even the nodes fairly distant from the target vertex v can increase the weights of the vertex v.

MCGO is more effective to detect non-essential proteins. For example, BZZ1 is a non-essential gene but most algorithms classify it as essential as the degree is 178. However, when

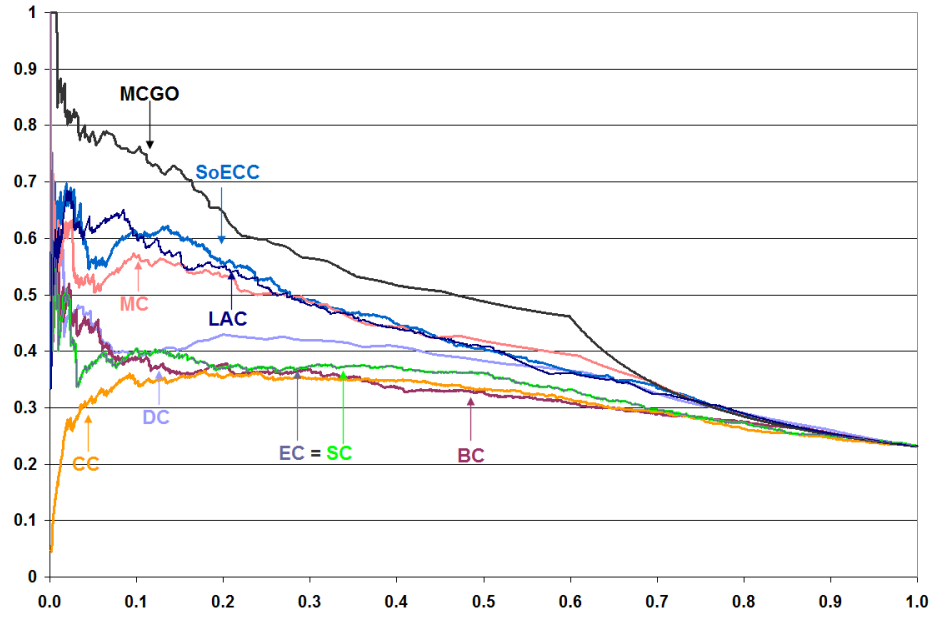


Figure 5.4: PR curves: MCGO is at the most upper-right-hand side indicating as the best algorithm.

EDGEGO is applied, the degree is reduced to 25, which in turn reduces the rank in MCGO, and even in other *-GO* algorithms. On the other hand, the degree of essential gene, RIO1, is reduced by only 6 which does not change the decision.

5.5 Summary and Future Work

A number of centrality algorithms have been used to discover essential proteins. However, all algorithms depend only on the structural properties, and they are sensitive to false links in a network. In this chapter, we show that the combination of network motifs and biological annotation improves the detection rates greatly, by proposing a new centrality algorithm, MCGO. MCGO uses network motifs for the detection of essential proteins in an edge-pruned network by EDGEGO, which trims edges based on GO terms. Due to motifs' statistical importance and because of its biological information, MCGO is more robust and biologically more meaningful.

Experimental results on a yeast PPI network show that MCGO improves significantly, compared to other existing algorithms of DC, BC, CC, SC, EC, SoECC and LAC algorithms in the three evaluation measures, which are 'TR proportion,' 'statistical measures' and 'precision-recall

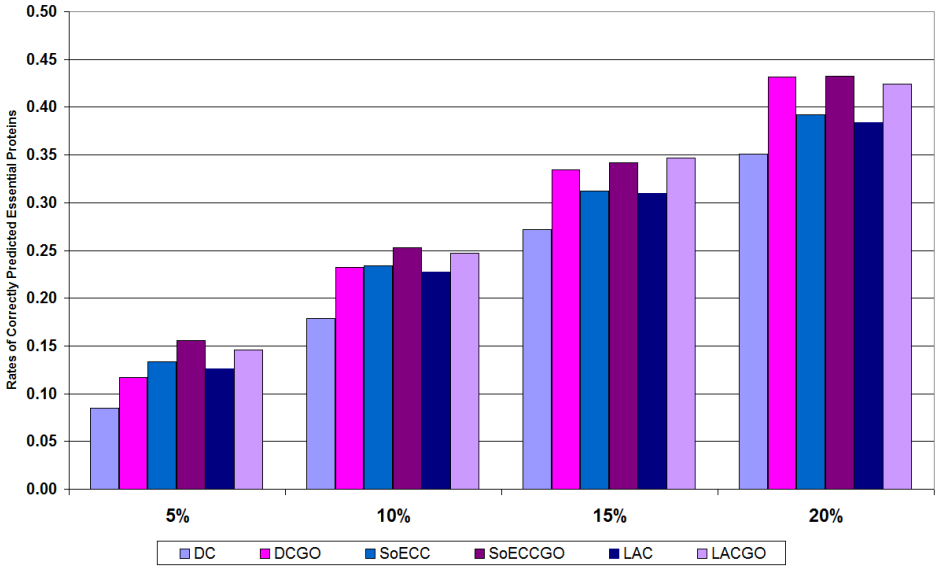


Figure 5.5: TR proportion: Each bar indicates the performance result of DC, DCGO, SoECC, SoECCGO, LAC, and LACGO from the left.

curve.' In addition, EDGEGO is applied to other algorithms, producing *-GO* algorithms, in order to examine the effect of EDGEGO. We observe that all *-GO* algorithms are much better than its original algorithms, however MCGO is still the best as it benefits from the robustness of network motifs.

The work has two contributions: 1) We use network motifs and GO for the discovery of essential proteins for the first time; 2) We show that pruning the network improves the performance significantly even with other algorithms. In near future, the algorithms should be examined to other organisms than a *Saccharomyces cerevisiae*. Furthermore, as an influential and robust measure for centrality, MC can be applied for different problems to other complex networks than PPI, such as gene regulatory networks or metabolic networks.

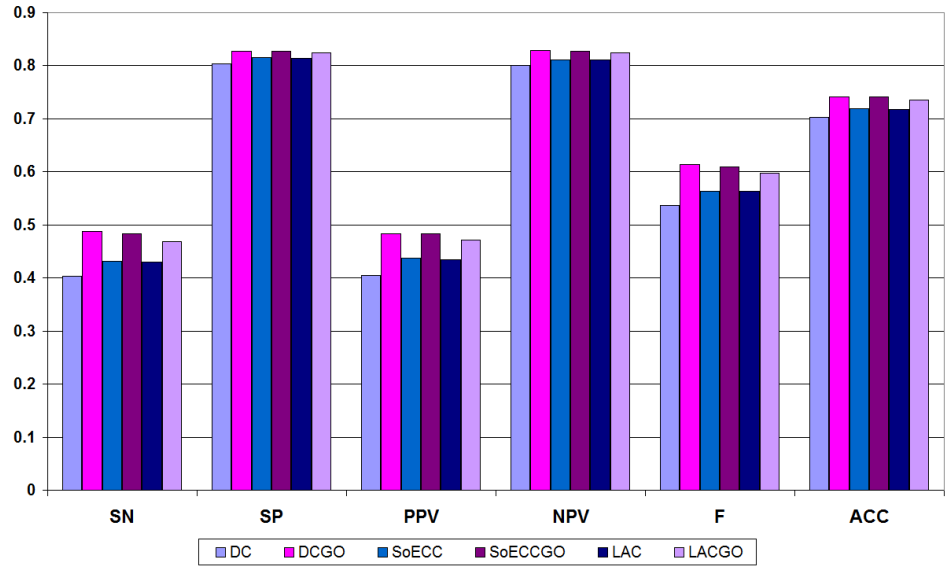


Figure 5.6: Statistical measures: Each bar indicates the performance result of DC, DCGO, SoECC, SoECCGO, LAC, and LACGO from the left.

Table 5.1: Confusion matrix or contingency table

	actual positive	actual negative
predicted positive	TP	FP
predicted negative	FN	TN

Table 5.2: Area under curve (AUC) value for each PR curve

Method	**AUC**	**Method**	**AUC**
DC	0.364	**DCGO**	0.457
BC	0.325	**BCGO**	0.397
CC	0.309	**CCGO**	0.367
SC	0.335	**SCGO**	0.425
EC	0.335	**ECGO**	0.420
SoECC	0.417	**SoECCGO**	0.478
LAC	0.413	**LACGO**	0.476
MC	0.410	**MCGO**	0.483

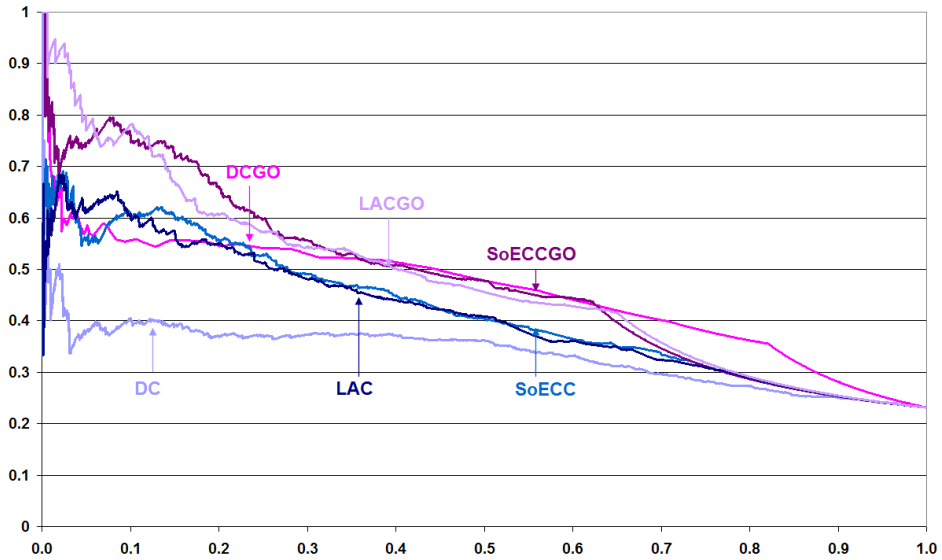


Figure 5.7: PR curves: Each *-GO* algorithm is better than its original algorithm.

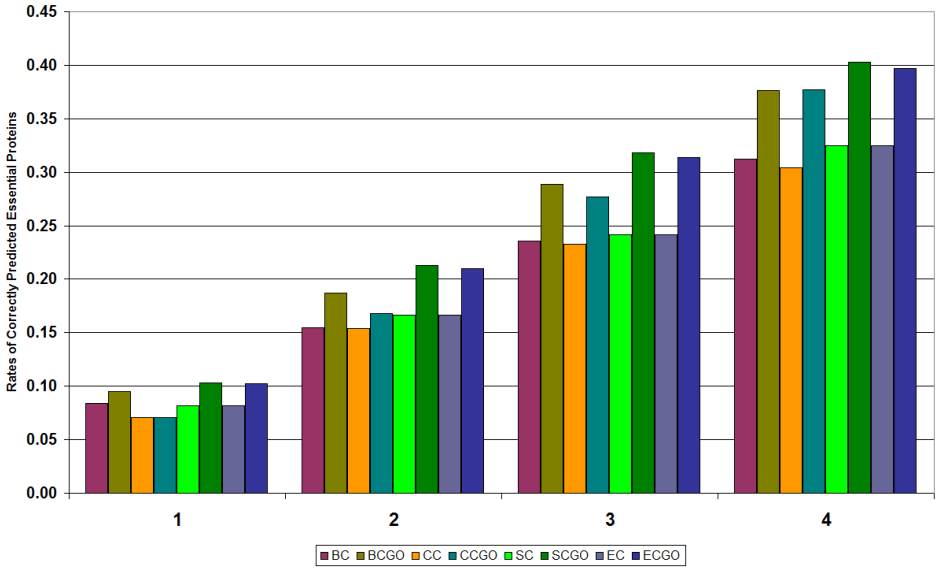


Figure 5.8: TR proportion: Each bar indicates the performance result of BC, BCGO, CC, CCGO, SC, SCGO, EC and ECGO from the left.

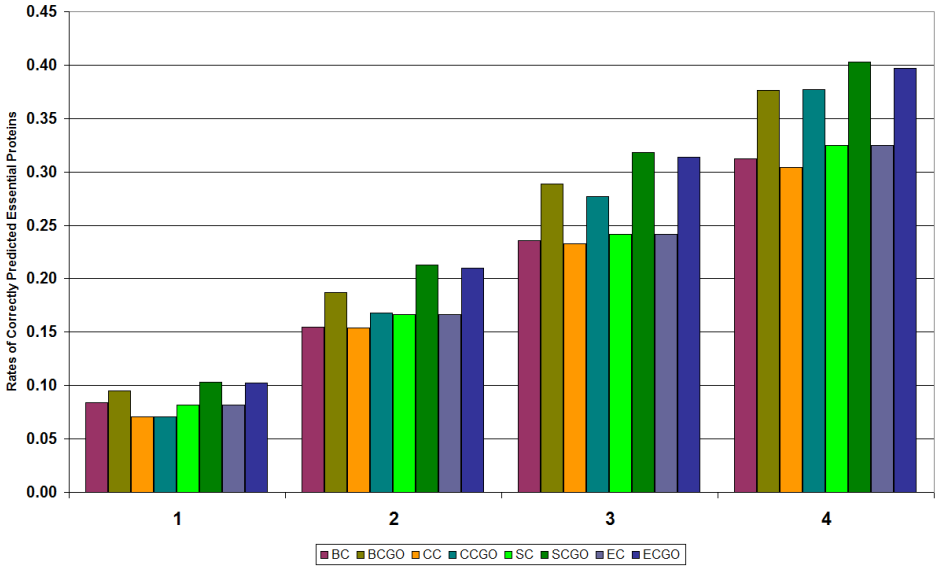


Figure 5.9: Statistical measures: Each bar indicates the performance result of BC, BCGO, CC, CCGO, SC, SCGO, EC and ECGO from the left.

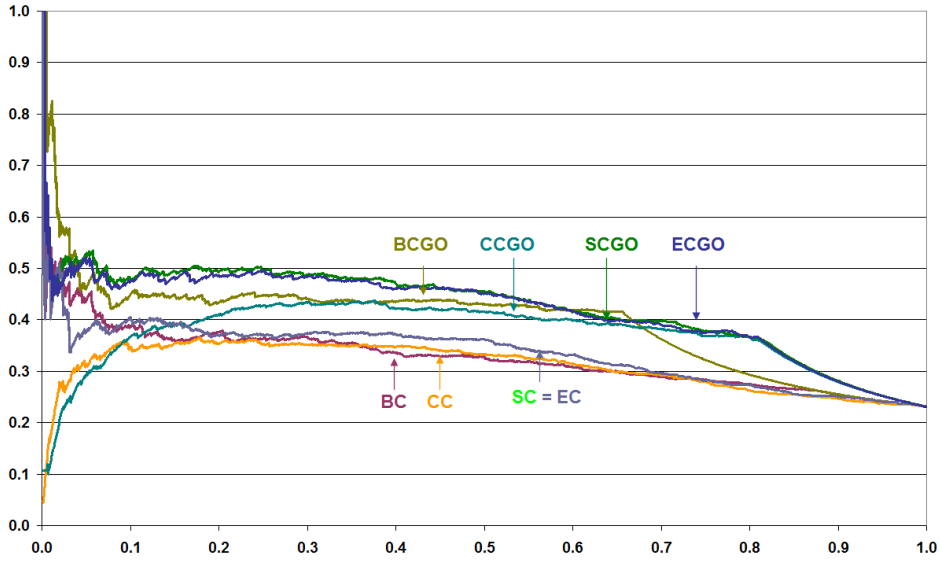


Figure 5.10: PR curves: Each -*GO* algorithm is better than its original algorithm.

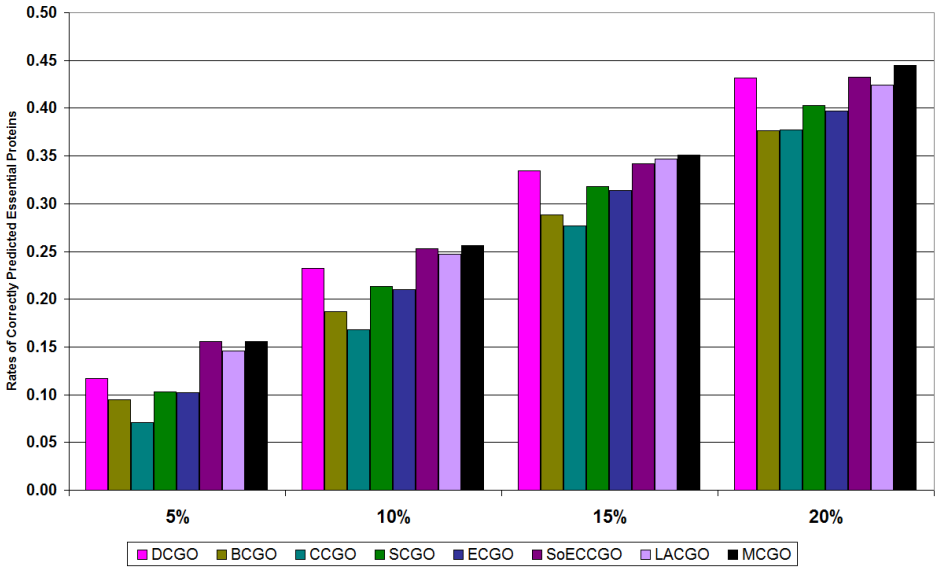


Figure 5.11: TR proportion: Each bar indicates DCGO, BCGO, CCGO, SCGO, ECGO, SoECCGO, LACGO and MCGO from the left.

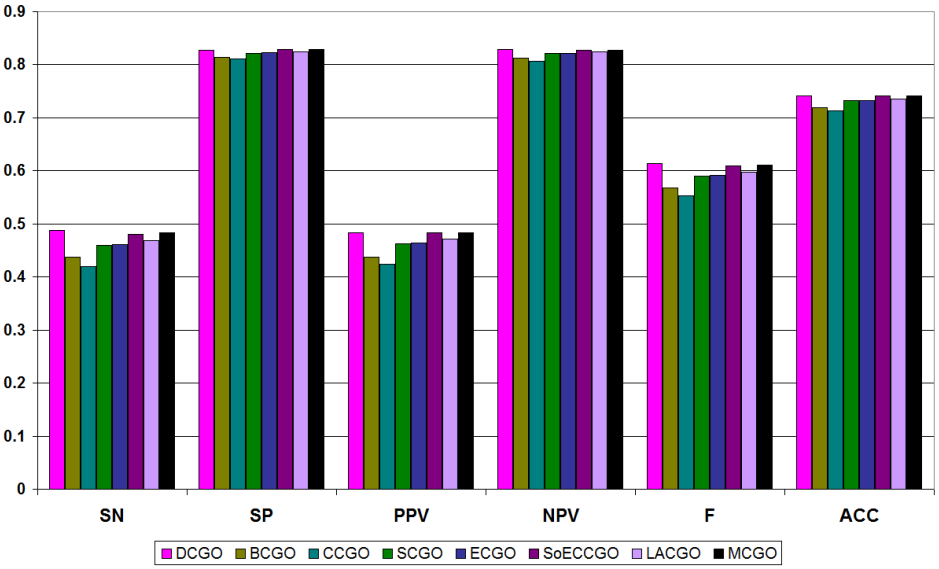


Figure 5.12: Statistical measures: Each bar indicates DCGO, BCGO, CCGO, SCGO, ECGO, SoECCGO, LACGO and MCGO from the left.

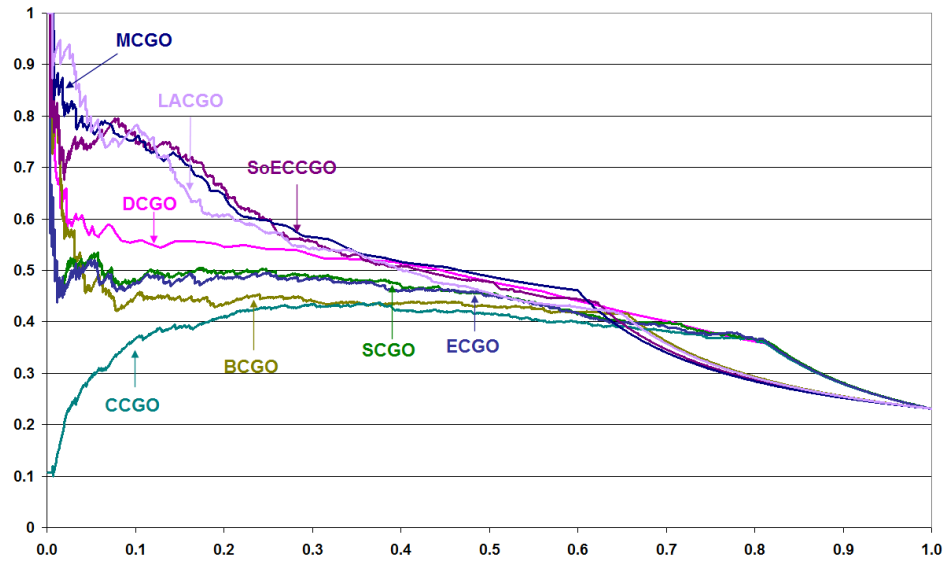
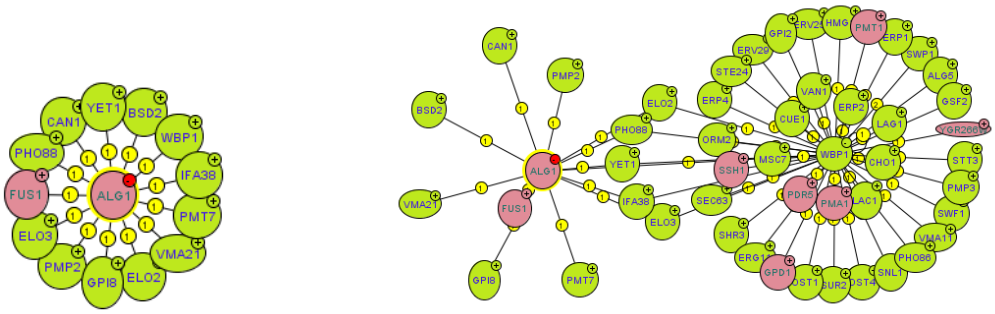


Figure 5.13: PR curves: The curve of MCGO is at the most upper-right-hand side.



(a) ALG1 and its direct neighbors

(b) WPB1 is extended with its neighbors and automatically some of other neighbors of ALG1 are connected to WPB1

Figure 5.14: The graphical view generated from the MINT web site [14] for ALG1 and their neighbor nodes. (b) shows the extended nodes which are neighbors of WBP1 in (a).

Chapter 6

MODEL-DRIVEN APPROACH TO PREDICTING ESSENTIAL PROTEINS IN A PPI NETWORK

6.1 Background

We reviewed in the previous chapter that essential proteins are important features for a cellular life in an organism and significant for drug design [241, 273] in practical usages. Therefore, identifying essential proteins has been a major research in genomic researches. Biological experiments such as single gene knockouts [243], RNA interference [244] and conditional knockouts [245] conventionally have detected essential proteins and stored them into databases. However, experimental screens are very time-consuming and resource-consuming. Therefore, various computational approaches which exploit essential protein databases have been recently developed. Essentiality in genes was typically discovered in a core conserved minimal set derived by comparison of multiple genomes as appeared in a number of researches [248, 249]. Alternatively, homology mapping techniques have been used to discover essential genes in newly sequenced organisms by searching similar sequences from databases [250]. These techniques require standard metric to measure similarity between target gene and the reference genes to determine the best matching.

On the other hands, it is observed that essentiality in genes is related to specific properties such as network centralities and hubs in biological networks. In protein-protein interaction (PPI) networks, which are undirected graphs with proteins as nodes and interactions as edges, proteins of densely connected hub nodes are known to be closely related to essential proteins [253]. This connection is called "centrality-lethality rule" and the relationship has been observed in many PPI networks with various centrality algorithms such as degree centrality (DC) [144, 255, 256], betweenness centrality (BC) [257, 258], closeness centrality (CC) [259], subgraph centrality (SC) [260] and eigenvector centrality (EC) [261]. The centrality algorithms can detect essential proteins greatly better than random selection [262], and their performances have been compared recently

[34, 254, 263]. However, detecting essential proteins based only on centrality scores might be insecure as current networks are still growing and incomplete [274] and most centrality algorithms are sensitive to the false links in networks.

Another computational studies focus on 'predicting' essential proteins rather than 'detecting', and they include Bayesian statistical approaches by Lamichhane *et al.* [275] and by Seringhaus *et al*. [251], or, machine learning techniques by Jeong *et al.* [276] and by Acencio and Lemke [35]. These studies are based on an essentiality model constructed with a number of features driven by their topological or biological properties. The features include specific functions in genes [275], characteristic sequence features [251], expression level fluctuation [276], topological properties and gene ontology terms [35]. Zotenko *et al.* [162] asserted that the essential proteins are more closely related with their Gene Ontology (GO) annotation terms than centralities in networks. In fact, Kim *et al* [34] recently showed that the involvement of GO into centrality algorithms improves the performances greatly.

6.2 Problem Statement

In this chapter, we want to design a model for better prediction of essential proteins, based on a set of features consisting network topology and biological information, so that we can improve the prediction rate for essential proteins in a PPI network, compared with existing models [35, 251, 275, 276].

Essential proteins have been detected through various centrality measures, or predicted based on various machine learning techniques as described in Chapter 5. Previous methods utilized topological properties or biological properties to collect a set of features that might determine the essentiality in proteins. In this work, we propose to build a model with the combination of topological and biological features in a protein-protein interaction (PPI) network, based on machine learning techniques. To show the performance improvement clearly, we provide two ways of feature combination in the following order.

First, we combine eight centrality measures as a set of features to plug into a machine learning classifier algorithm, and name the set of features as a ***CENT-GO*** (Kim et al., 2012). In the

construction of CENT-GO, we incorporate biological information using gene ontology terms (GO) so that they play biologically informative as well as topologically valuable roles. The meta classifier used in this test is based on seven decision trees, support vector machine and neural network algorithms. We compare the results with those by Acencio and Lemke [35]. They extract total 23 features, which we call as ***ING-GO*** (Acencio and Lemke, 2009), including 12 topological properties derived from an integrated network and 11 gene ontology terms.

Next, we combine CENT-GO and ING-GO to create a ***CENT-ING-GO*** feature set including total 31 features. The CENT-ING-GO feature set is compared with CENT-GO and ING-GO with the same classifier. Experimental results show that CENT-ING-GO, although computationally more expensive, is the most effective feature set in the given data. Additionally, we confirm that the improvement of the data with CENT-ING-GO over CENT-GO or ING-GO is statistically significant by Mann-Whitney U-statistics [277] test.

Additionally we analyze each individual feature with one-feature data and a rule generation method based on a decision tree algorithm. In this way, we could see that the impact of each individual feature on the essentiality as well as the significantly improved impact when they are integrated together.

6.3 Methods

In this study, we extract eight scores for each protein from a PPI network, downloaded from BioGRID [278] server. To incorporate biological information with each feature, gene ontology (GO) annotation terms are used to refine the PPI network using EDGEGO algorithm. In addition, we combine the additional features obtained from the authors in [35] to see the improved performance.

6.3.1 Algorithms

We incorporate various centrality measures with GO annotation terms to extract valuable features from our data set.

6.3.1.1 Centrality Measures

Centrality measures have been used to determine more influential individuals from a social group [13] in social networks, and they were applied to analyze biological networks to predict essential proteins in PPI networks [34, 254, 263, 265, 266] or to detect global gene regulator in gene regulation networks [267]. However, the centrality is highly dependable on different application contexts, as depicted in Figure 5.1. Therefore, various centrality measures are developed with different interpretations for different purposes [34, 144, 254–261, 263], such as, degree centrality (DC), betweenness centrality (BC), closeness centrality (CC), subgraph centrality (SC), eigenvector centrality (EC), sum of edge clustering coefficient centrality (SoECC), local average connectivity (LAC), and motif centrality (MC) measures. Recent studies [34, 254, 263] compared these centrality measures for the task of identifying essential proteins in a PPI network. In this work, we create a set of features with the eight centrality measures of DC, BC, CC, SC, EC, SoECC, LAC and MC. Detailed formulation for each centrality measure is reviewed in Chapter 5.

6.3.1.2 Gene Ontology and EDGEGO algorithm

To make the centrality features biologically significant, we first use EDGEGO algorithm introduced in Chapter 5 as Algorithm 5 to remove a number of 'biologically insignificant' edges from the PPI network. In this algorithm, biologically insignificant edges are determined with Gene ontology (GO) [231] terms associated with it. GO terms, providing annotations of gene and gene product attributes across species and databases, consist of three independent domains: biological process (BP), molecular function (MF) and cellular component (CC). With the three orthogonal aspects as roots, GO is represented as a directed acyclic graph (GO DAG) in Figure 4.2. GO DAG describes each GO term as a node and the relationships as an directed edge with hierarchical structure, where children are more specific than the parents. In GO DAG, if a gene ge is annotated with a GO term pe, then ge is also annotated with all the ancestors of pe. With the root as depth 0, hence, the depth of a GO term represents its information depth as well. EDGEGO algorithm, shown in the Algorithm 5, removes a number of edges given the threshold of information depth,

and as a result, less informative edges are removed in the graph. In this way, biological information is incorporated into the network and computational cost is greatly reduced because of the reduced edge sizes in a network.

We first apply EDGEGO to an yeast PPI network to obtain a GO-pruned PPI, then compute eight centrality measures for each protein. We name the set of features as CENT-GO which includes DCGO, BCGO, CCGO, SCGO, ECGO, SoECCGO, LACGO and MCGO. In most cases, the number of essential proteins and non-essential proteins are imbalanced, that is, the number of non-essential proteins tend to be larger than that of essential proteins. As the imbalanced data sets usually degrade the prediction quality in machine learning algorithms [279], we undersample the non-essential proteins to obtain several balanced data sets to have the same number of essential and nonessential proteins in each data set. Figure 6.1 visualizes the preprocessing.

We compare the performance of CENT-GO (Kim et al., 2012) with ING-GO (Acencio and Lemke, 2009) [35] which consists of 23 features, from which 12 are topological features from an integrated network (INGI) and 11 are GO annotation terms. The integrated network INGI combines a PPI, a transcriptional regulatory (TRN) and a metabolic network of an yeast. 12 topological features include DC in PPI network, in- and out-degree in metabolic network, in- and out-degree in TRN, clustering coefficient, BC in the integrated network, BC in PPI network, BC in TRN, BC in metabolic network, CC in integrated network and identicalness (the number of genes with identical network topological characteristics). 11 biological features consist of five cellular localization annotation of “cytoplasm”, “endoplasmic reticulum”, “mitochondrion”, “nucleus” and “other localization”, and six biological process annotation of “cell cycle”, “metabolic process”, “signal transduction”, “transcription”, “transport” and “other process.” Figure 6.2 summarizes the data set preparation process of “ING-GO,” which also results in a number of balanced data sets.

6.3.1.3 Classifier

In a number of previous studies [34, 254, 263], each centrality measure gives a score for each protein then the proteins with relatively high scores are assumed as essential proteins. In this study, however, we want to focus more on systematical approach for predicting essential proteins,

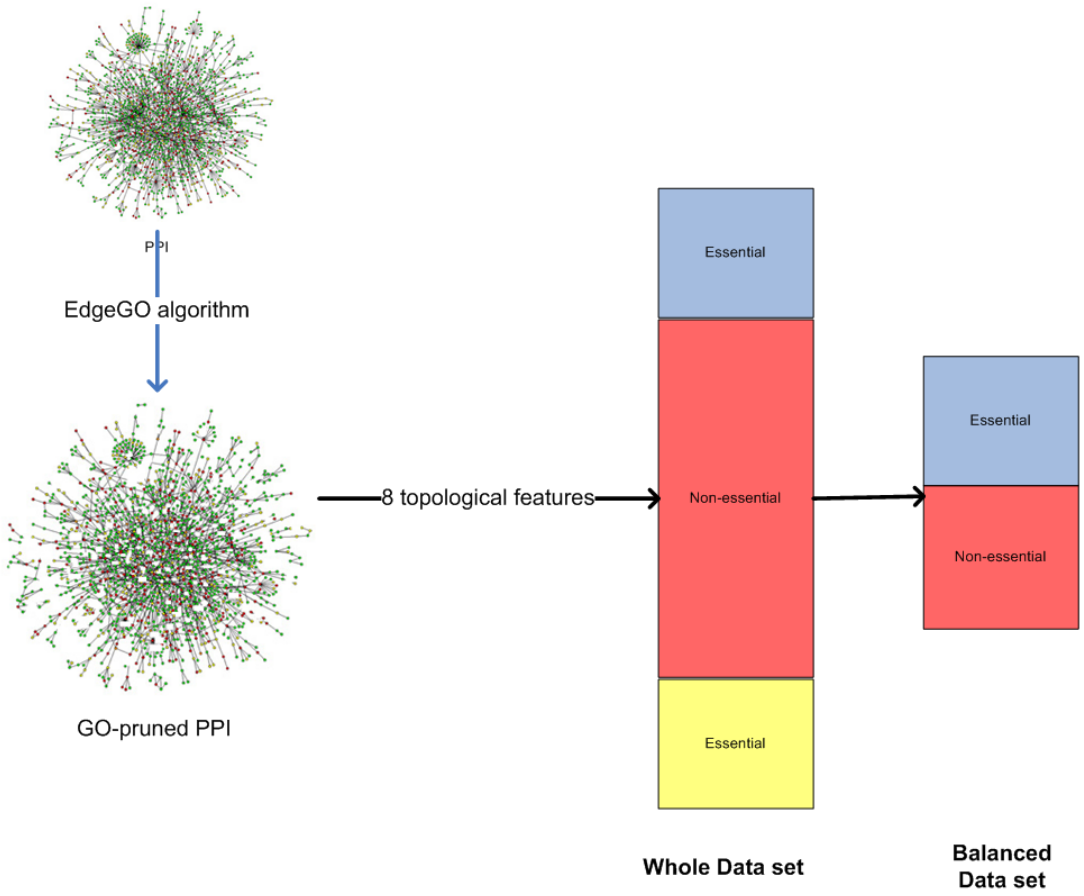


Figure 6.1: The process of CENT-GO extraction based on centrality measures from GO-pruned PPI network: In the left, an yeast PPI network is pruned with EDGEGO algorithm, where 14,925 interactions are removed out of 37,209 interactions total. For each vertex, eight centrality measures are calculated from the GO-pruned PPI, each of which is a feature of the protein node. The imbalanced data set is under-sampled to form a balanced data set.

by incorporating biological information with centrality measures. Therefore, we take a model-driven approach using machine learning techniques: We extract a set of features, then design a classifier to categorize data and evaluate the performance through some validation methods. Here, we utilize WEKA (Waikato Environment for Knowledge Analysis) package [280] for classifying and evaluating the results. Figure 6.3 is the architecture of a classifier we designed. We ensemble nine learning algorithms, using "Vote" method [281] which combines the outputs of each classifier with various rules, and we select an average rule. The nine algorithms include seven decision-tree based algorithms of (1) REPtree [282], (2) random tree [282], (3) random forest [282, 283], (4) C4.5 (J48) [282, 284] with at least 29 instances per leaf, (5) best first tree (BFT) [282, 285] with at

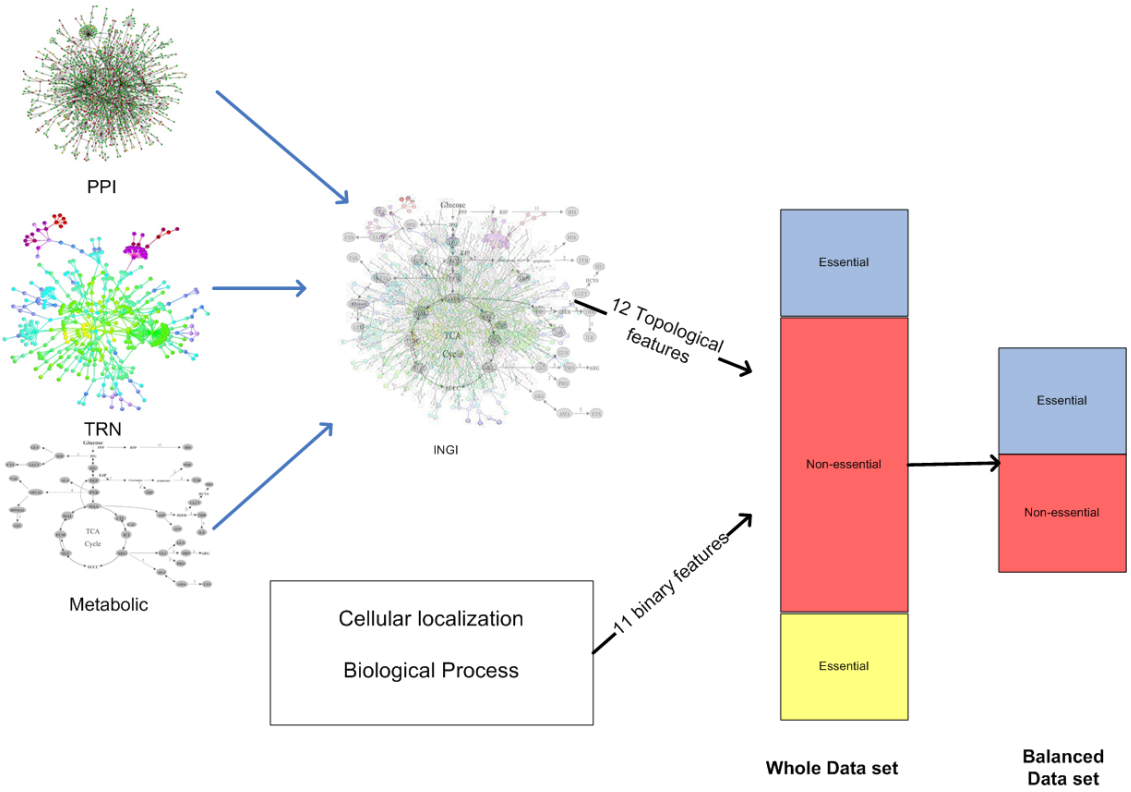


Figure 6.2: The process of ING-GO extraction based on an integrated network and BP GO and CC GO terms: An yeast PPI, transcriptional regulatory and metabolic network is integrated into an integrated (INGI) network. 12 topological features are extracted from this INGI and 11 features are obtained from biological process and cellular localization GO terms. Each protein consists of 23 features and a balanced data set is also obtained with undersampling.

least 26 instances per leaf, (6) logistic model tree (LMT) [282,286] and (7) alternating decision tree (ADT) [282, 287], and a support vector machine (SVM) [288] with RBF kernel and standardizing filter and multi-layer perception or neural network [289] algorithm. Here, a "bagging" algorithm is applied to each algorithm before merging them, in order to reduce variances.

6.3.2 Evaluation Methods

The meta-classifier is applied to the balanced data set with CENT-GO (Kim et al. 2012), ING-GO (Acencio and Lemke, 2009) or the combination of two, CENT-ING-GO features. For validation, we used 10-cross folding technique. For each test, total 10 runs of classification and evaluation is performed and the result is the average of 10 runs. At each run, the data set is divided

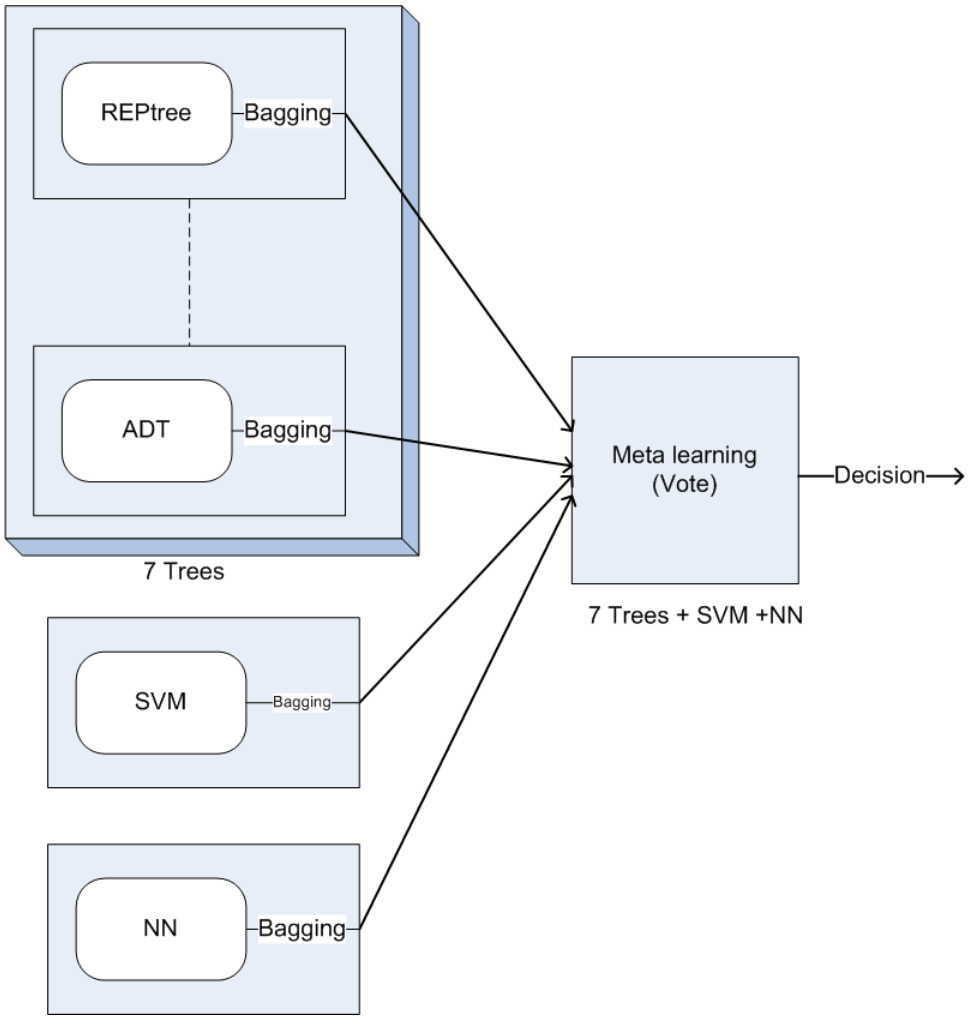


Figure 6.3: *Classifier by Kim et al, 2012*: 7 decision-tree based algorithms, a support vector machine (SVM) and neural network method, to each of which 'bagging' is applied for variance reduce, are combined into a meta-classifier.

into a training set and a testing set and the classifier builds a model based on each training set then perform prediction on the testing set. As the task is basically a binary decision problem, Receiver Operator Characteristic (ROC) or Precision-Recall (PR) curves are appropriate for assessing the performances. PR curve is an alternate method over ROC in the skewed data [270–272], but ROC is more popular method if the data is evenly distributed. Since we use a balanced data set, we do not need to constraint in the PR curve. Therefore, we provide both of the ROC and PR results with their area-under-curve values (AUC) as well as an accuracy at an optimal threshold (ACC). We also provide the computational time (T) to compare the time efficiency.

The performance was compared not only based on different feature sets but also based on different classifiers. We compared the performance of our classifier of Figure 6.3 against a different classifier used in [35] , which is the combination of 8 decision-tree-based methods, shown in Figure 6.4. This classifier combines total 8 decision tree algorithms, where a naive Bayes tree (NBT) [290] and the 7 decision trees in our classifier. More detail setting information is provided as a supplement document in [35]. For the sake of clarity, the 8 decision-based classifier is noted as *Classifier by Acencio and Lemke, 2009* and ours as *Classifier by Kim et al, 2012*.

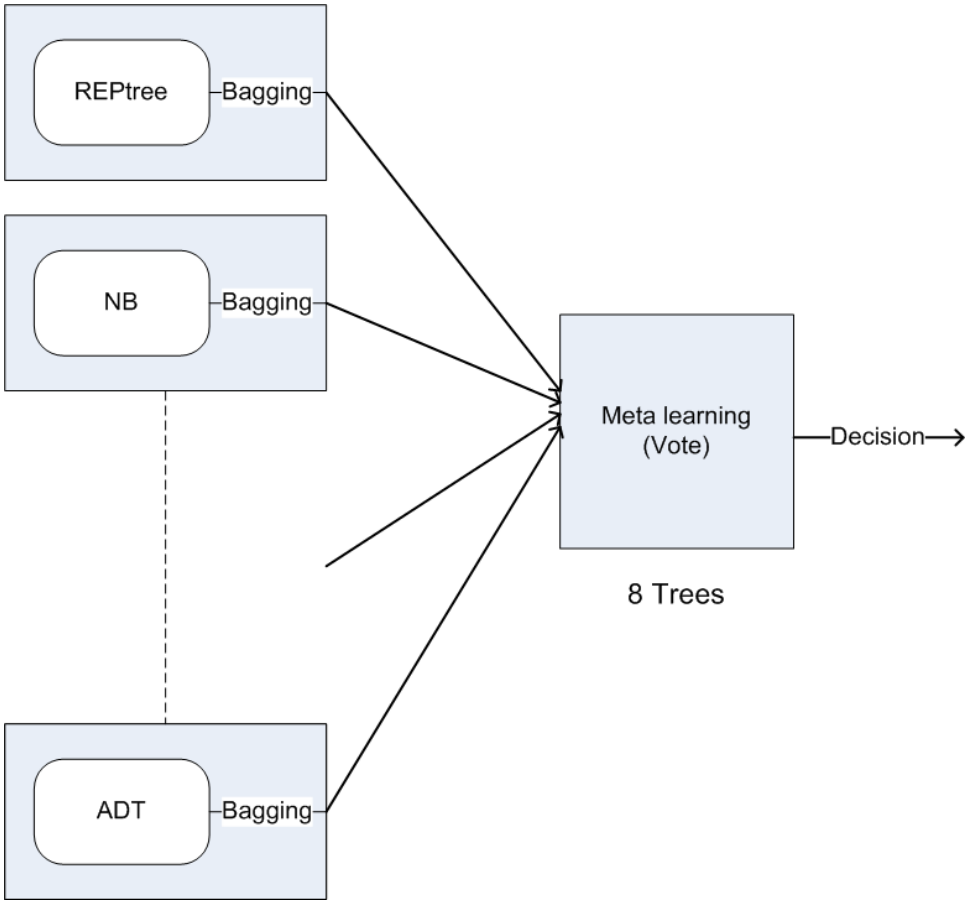


Figure 6.4: *Classifier by Acencio and Lemke, 2009* : 8 decision-tree based algorithms to each of which 'bagging' is applied for variance reduce, are combined into a meta-classifier.

In addition, we used the StAR (Statistical Analysis of ROC curves) [291] which is an available tool in a web (http://protein.bio.puc.cl/cardex/servers/roc/home.php) to determine if a result is significantly better than others, based on Mann-Whitney U-statistics test.

6.4 Results and Discussion

6.4.1 Data sets and features

Previously, Acencio and Lemke [35] used a decision-tree-based meta classifier (*Classifier by Acencio and Lemke, 2009*) to predict essential genes using a combination of topological features and gene ontology terms. Twelve topological features were extracted from an integrated network of gene interactions (INGI) which combines PPI, a gene regulatory network and a metabolic network of *Saccharomyces cerevisiae*. The INGI contains 5,667 genes with 72,806 interactions and 96% of genes are protein-coding genes. The 5,667 genes consist of 1,024 essential genes, 4,097 non-essential genes and 546 unknown genes. To provide additional information besides the topological features of the genes, the authors [35] added 11 cellular localization and biological process information to each gene, which is simply a true or false based on its annotation. With a decision-tree based meta classifier on the data set of 23 features, the authors provided various experimental results and compared the performances based on each feature or different combination of the features.

The purpose of our study is to predict *essential proteins*, rather than *essential genes*. With this goal, we extract features only from a PPI of S. cerevisiae (yeast) downloaded from a BioGRID database [278], which is the same PPI as in the study by Acencio and Lemke [35]. The PPI includes 37,209 interactions and 4,854 proteins, where 998 are essential, 3,557 are non-essential and others are unknown proteins. To extract the features, first, we filter out a number of interactions in the PPI network using EDGEGO algorithm which removes relatively uninformative edges using GO terms, and obtain a reduced network. We name it as a ***GO-pruned PPI***. Then we compute eight centrality measures in the GO-pruned PPI and name them DCGO, BCGO, CCGO, SCGO, ECGO, SoECCGO, LACGO and MCGO.

6.4.2 Comparison of the balanced data sets

The data set includes 998 essential proteins and 3,557 non-essential proteins, which is an imbalanced data set where the ratio of positive and negative data sets are unbalanced. In ma-

chine learning, the prediction is greatly biased to the majority class, as it was discussed in the review [279]. Hence we construct balanced data sets by undersampling non-essential proteins, which follows the technique introduced in the study [35]. We make 10 balanced data sets, each of which contains all the 998 essential proteins and the same number of non-essential proteins randomly selected from 3,557 non-essential proteins. We analyze the data sets with 2-fold cross validation classification method with the *Classifier by Kim et al., 2012*, on each data set with CENT-ING-GO feature set. To verify that the balanced data sets are statistically similar, we used Mann-Whitney U-statistic [277] to compare the results based on the area under ROC (AUC-ROC) values. In the Mann-Whitney U-statistic test, if any two experiments produce significantly different performances, then the P-value will be smaller than a threshold which is usually 0.05. Figure 6.5 shows the ROC curves and Table 6.1 shows the P-value of each pair of data sets. The ROC curves are very similar with a range of $.80 \sim .81$ AUC-ROC and the U-statistics test verifies that all 10 data sets have similar performances as all the P-values are higher than 0.05 as demonstrated in Table 6.1. After confirming the similar performances among balanced data sets, we randomly choose one data set for further experiments.

Table 6.1: Comparison of balanced data sets: To verify that all the 10 data sets are statistically similar, we run a meta classifier to each data set and obtain an AUC-ROC value. We verified that all the data sets are statistically similar through Mann-Whitney U-statistics test with their AUC-ROC values.

	data0	**data1**	**data2**	**data3**	**data4**	**data5**	**data6**	**data7**	**data8**
data1	0.9945								
data2	0.7708	0.4235							
data3	0.9828	0.9407	0.4858						
data4	0.8233	0.8600	0.8661	0.8884					
data5	0.9120	0.7860	0.3362	0.7275	0.7702				
data6	0.9013	0.7064	0.6592	0.7920	0.9803	0.5383			
data7	0.9474	0.8448	0.5427	0.9080	0.9289	0.6678	0.8804		
data8	0.8885	0.7019	0.6857	0.7571	0.9949	0.5416	0.9661	0.8317	
data9	0.8847	0.6948	0.2729	0.6492	0.7403	0.9266	0.4754	0.5870	0.4769

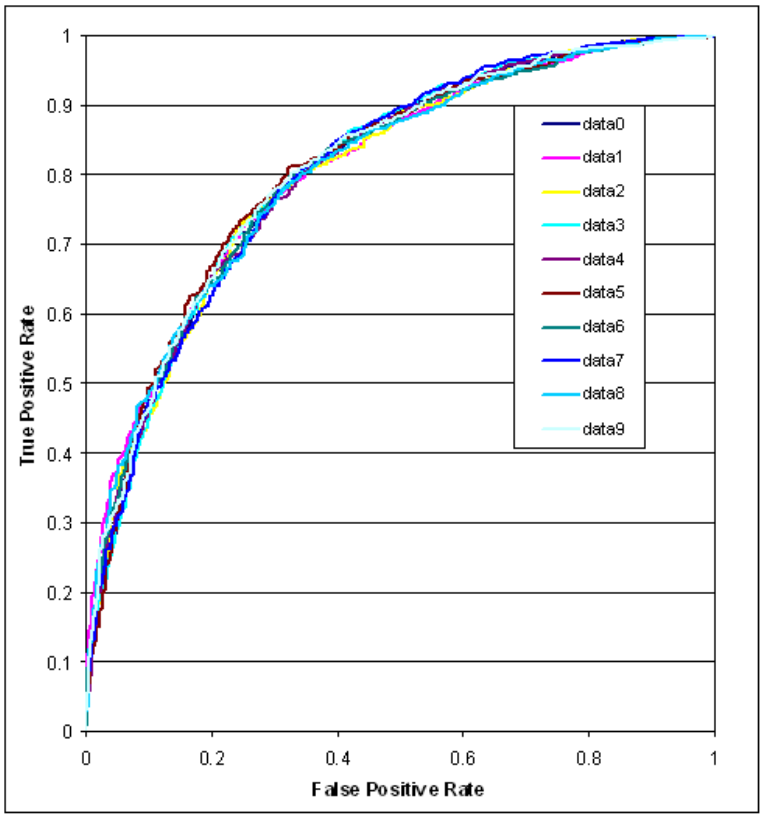


Figure 6.5: ROC curves of the ten balanced data sets with CENT-ING-GO features: All 10 data sets have similar performances.

6.4.3 CENT-GO and CENT-ING-GO

In our experiments, we choose one balanced data set with CENT-GO and analyze it with 10-fold classification method. The performances are compared based on the four evaluation measures; area under ROC (AUC-ROC), area under PR (AUC-PR), accuracy rate at the optimal threshold (ACC) and the computational time (T) of model building. The classification with CENT-GO performs significantly better than each individual centrality feature, as shown in Figure 6.6, Figure 6.7 and Table 6.6, with the AUC-ROC as .784, AUC-PR as .781 and ACC as .727. Table 6.5 confirms that the performance with CENT-GO is significantly better than that of each single feature.

Next, we compare the results of CENT-GO (Kim et al., 2012) with ING-GO (Acencio and Lemke, 2009) [35]. Table 6.3 indicates that CENT-GO is better than ING-GO in terms of ACC and computational time. However, ING-GO performs better in terms of AUC-ROC and AUC-PR.

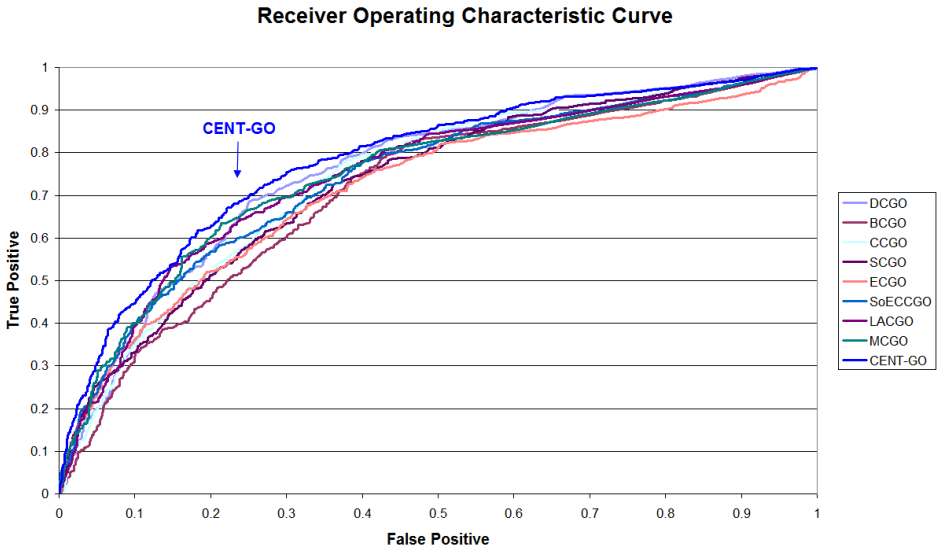


Figure 6.6: ROC curves for individual feature sets and CENT-GO: The prediction with CENT-GO performs significantly better than with each individual measure. We also notice that DCGO, MCGO, LACGO, and SoECCGO shows relatively good scores which are characterized as local features.

But, we should note that Mann-Whitney U-statistic [277] test shows that the two performances are not significantly different in AUC-ROC as shown in the Table 6.2.

Nextly, to see a significant improvement, we combined the CENT-GO and ING-GO features to make a *CENT-ING-GO* feature set and run the *Classifier by Kim et al, 2012* to the data of CENT-ING-GO. As a result, we were able to obtain a significantly improved result as shown in Table 6.3 with increased AUC-ROC (.818), AUC-PR(.804) and ACC (.753), although it is computationally more costly. Also the Mann-Whitney U-statistic [277] test verifies that the improvement of CENT-ING-GO is statistically significant, compared with each of CENT-GO and ING-GO. The improvement is also visualized with the ROC curves in Figure 6.8 and PR curves in Figure 6.9.

6.4.4 Prediction based on different classifiers

The classifier used in this chapter is a meta classifier based on seven decision-trees, support vector machine and neural network algorithm. Each algorithm is first applied with "Bagging" method to reduce variance, then they are combined into a classifier by averaging their probability

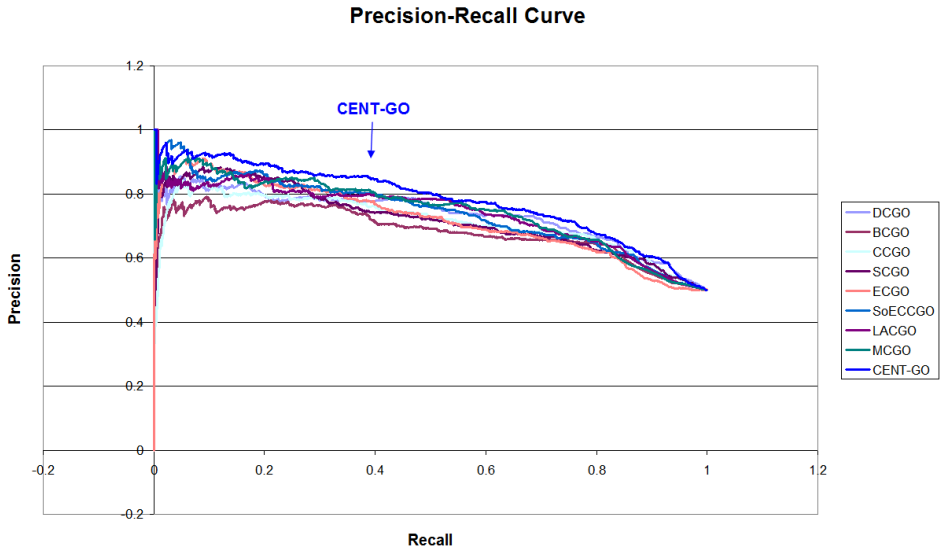


Figure 6.7: PR curves for individual feature sets and CENT-GO: The prediction with CENT-GO performs significantly better than with each individual measure. We also notice that DCGO, MCGO, LACGO, and SoECCGO show relatively good scores which are characterized as local features.

Table 6.2: Statistical Significant of the set of integrated features: Each set of features was assessed based on its statistical significance. We can observe that when we run a classifier to the set of integrated features (CENT-ING-GO), the performance improves significantly.

	CENT-GO (Kim et al., 2012)	ING-GO (Acencio and Lemke, 2009)
ING-GO (Acencio and Lemke, 2009)	0.224	
CENT-ING-GO (CENT-GO + ING-GO)	**7.01E-06**	**2.11E-04**

estimates using "Vote" technology. Compared with the classifier used in [35] which is a meta classifier based on eight decision-trees, we could observe that the performance improved slightly. As the results are shown in Table 6.4, Figure 6.10 and Figure 6.11, "Classifier by Kim et al., 2012" performs better than "Classifier by Acencio and Lemke, 2009", although the difference is not statistically significant with the P-value as 0.31.

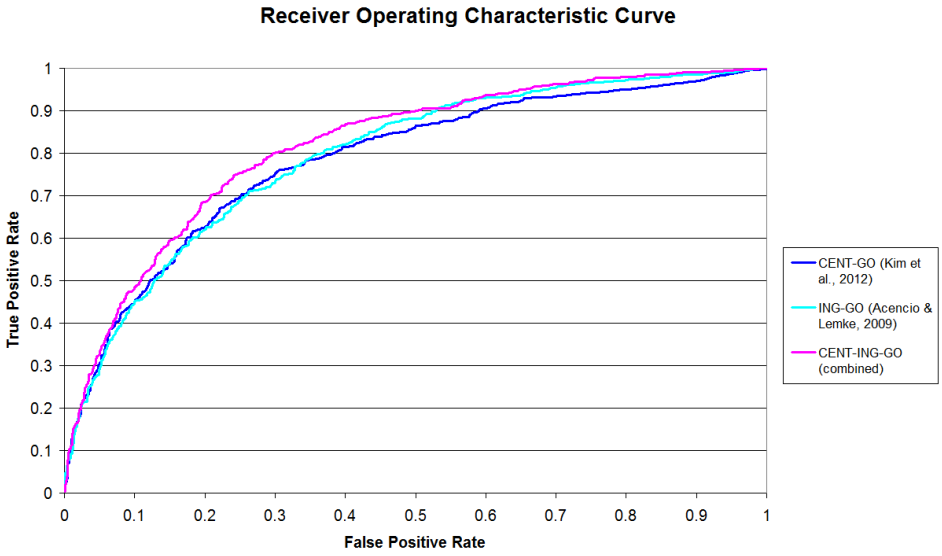


Figure 6.8: ROC curves of ING-GO (Kim et al, 2012), CENT-GO (Acencio and Lemke, 2009) and the CENT-ING-GO (CENT-GO + ING-GO): The prediction performance improves significantly with the integral of the two sets of features.

Table 6.3: The performances of CENT-GO, ING-GO and the combined of the two, CENT-ING-GO are compared with their area under ROC (AUC-ROC), area under PR (AUC-PR), accuracy (ACC) and time (T). CENT-GO and ING-GO have slight variations, but integration of them, CENT-ING-GO, can improve the performance significantly.

	AUC-ROC	AUC-PR	ACC	T
CENT-GO (Kim et al. 2012)	0.784	0.781	0.727	174.19
ING-GO (Acencio and Lemke, 2009)	0.793	0.784	0.723	446.24
CENT-ING-GO (CENT-GO + ING-GO)	0.818	0.804	0.753	620.79

6.4.5 Analysis on each centrality measure

To see the effect of an individual measure on essentiality determination, we analyze the results of each measure with a classifier. Table 6.6 indicates that the experiment with all the measures (CENT-GO) outperform the one with each individual measure, but, individually, DCGO has relatively better result than others; BCGO and ECGO are relatively poor measures. Table 6.5 compares the performance statistics on AUC-ROC values; The experiment with CENT-GO is statistically significant than all other experiments in the table. BCGO and ECGO are statistically inferior than

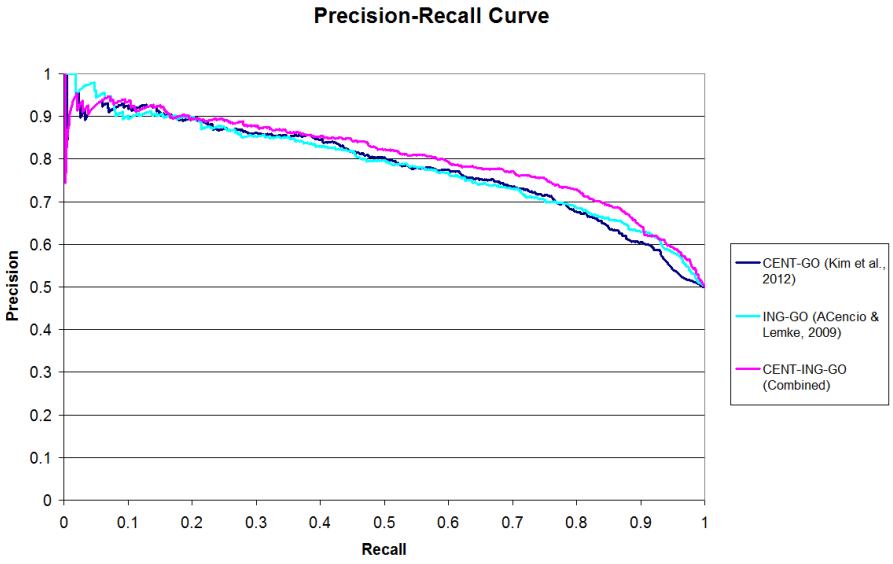


Figure 6.9: PR curves of ING-GO (Kim et al, 2012), CENT-GO (Acencio and Lemke, 2009) and the CENT-ING-GO (CENT-GO + ING-GO): The prediction performance improves significantly with the integral of the two sets of features.

Table 6.4: Comparison of classifiers: The CENT-ING-GO feature set is performed with 2 different classifiers and the performances are compared based on AUC-ROC, AUC-PR, ACC and T measures. "Classifier by Kim et al., 2012" performs better than "Classifier by Acencio and Lemke, 2009."

	AUC-ROC	AUC-PR	ACC	T
Classifier by Kim et al., 2012	0.818	0.804	0.753	620.79
Classifier by Acencio and Lemke, 2009	0.812	0.801	0.742	199.35

most of other measures. Additionally, we could see that the performance of MCGO and SoECCGO is closely similar with 0.977 of P-value.

We also analyze the features with a decision tree algorithm. Decision-tree is especially useful to discover classification rules as a tree structure. A representative decision-tree called *J48* implementing C4.5 algorithm [284] is used to generate a rule. We perform the algorithm to all ten balanced data sets, with two setting. One is for the CENT-GO features only, and the other for the CENT-ING-GO. Examining all ten balanced data sets, we obtained MCGO as a root for 6 data sets and DCGO as a root for 4 data sets, for both settings. In fact, it is interesting as MCGO is not better than DCGO in terms of AUC-ROC, shown in the Table 6.6. However this result is consistent with

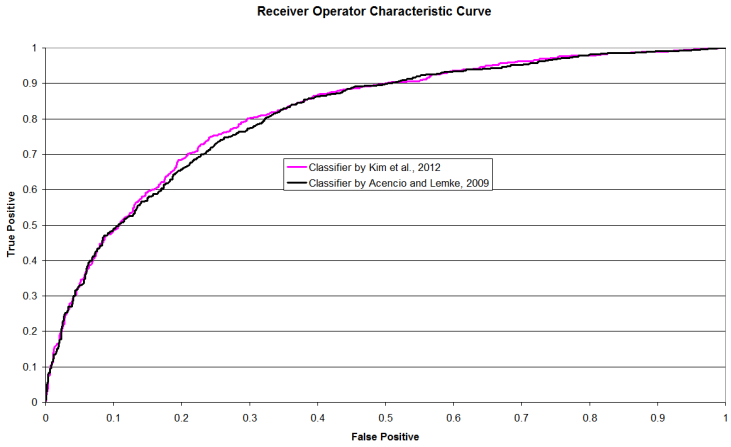


Figure 6.10: ROC curves for the two classifiers are provided. Classifier by Kim et al., 2012 is slightly better than Classifier by Acencio and Lemke, 2009.

the result of previous study in [34], where MCGO identifies more essential proteins than other centrality measures in PPI network. Figure 6.12 and Figure 6.13 are the decision-tree structure obtained with CENT-GO and CENT-ING-GO from dataset5, respectively. As shown in Figure 6.13, cellular localization features such as “nucleus” and “ endoplasmic reticulum” play important roles to determine essential proteins in most data sets, which is also consistent with previous study in [35] which concluded that cellular component is important for gene essentiality.

6.5 Summary and Future Works

Essential proteins play a critical role on the survival of organism, so the identification of essential proteins has been conducted through experimental or computational approaches. A number of centrality measures have been used to discover essential proteins as computational efforts. These measures which are originally dependent only on the structural properties in a network, can be incorporated with biological information for better performance. Acencio and Lemke [35] designed a machine learning based approach with topological properties and gene ontology terms to predict essential genes. Kim et al. [34] ranked each protein with motif centrality (MC) computed in a GO-pruned PPI network to detect essential proteins. In this work, we make use of existing

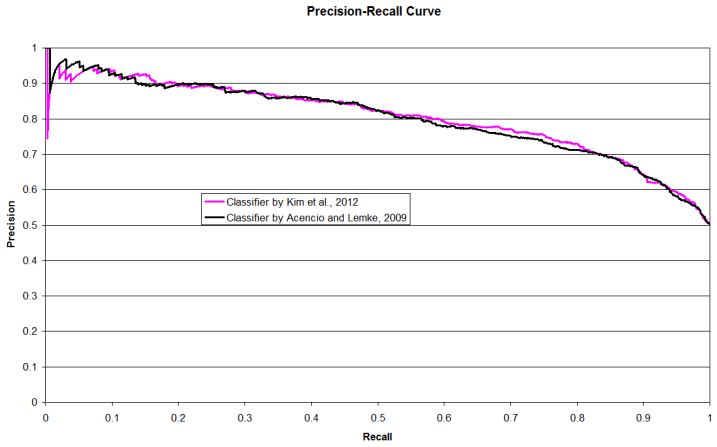


Figure 6.11: PR curves for the two classifiers are provided. Classifier by Kim et al., 2012 is slightly better than Classifier by Acencio and Lemke, 2009.

centrality measures and biological information to predict essential proteins with machine learning techniques in the yeast *Saccharomyces cerevisiae* PPI network.

We use a PPI network downloaded from the BioGRID database [278] with 4,854 proteins and 37,209 interactions. Out of 4,854 proteins, 998 are essential, 3,557 are non-essential and the rest are unknown. The network is first pruned by EDGEGO algorithm which removes 14,925 interactions of relatively uninformative GO terms. From the GO-pruned PPI network, we compute 8 centrality measures, namely, DCGO, BCGO, CCGO, ECGO, SCGO, SoECCGO, LACGO and MCGO. We name the set of 8 features as CENT-GO. Then we construct ten balanced data sets where the number of essential proteins and the number of non-essential proteins are the same, to avoid biased performance to the majority set. For evaluation measures, we used the area under ROC (AUC-ROC), area under PR (AUC-PR), accuracy at an optimal threshold (ACC) and computational time (T). We first confirmed that the 10 balanced data sets are statistically similar through Mann-Whitney U-statistics test on AUC-ROC, so that we can randomly choose one data set for further experiments.

We demonstrate that the prediction with CENT-GO has .784 for AUC-ROC, .781 for AUC-PR and .727 accuracy. The performance is compared with the 23 features of ING-GO (Acencio and Lemke, 2009) set used in [35]. ING-GO consists of 12 topological features extracted from

Table 6.5: Statistical Significance of each feature: Each feature was assessed based on its statistical significance. The performance improvement of CENT-GO is statistically verified as all the p-values, compared with each measure, are less than 0.05.

	CENT-GO	DCGO	BCGO	CCGO	SCGO	ECGO	LACGO	SoECCGO
DCGO	0.001							
BCGO	0.000	0.000						
CCGO	0.000	0.019	0.000					
SCGO	0.000	0.000	0.009	0.220				
ECGO	0.000	0.000	0.472	0.020	0.131			
LACGO	0.000	0.047	0.000	0.884	0.170	0.003		
SoECCGO	0.000	0.214	0.000	0.393	0.031	0.000	0.405	
MCGO	0.000	0.260	0.000	0.468	0.073	0.000	0.488	0.977

an integrated (PPI, transcriptional regulatory and metabolic) network and 11 biological process and cellular localization gene ontology features. With only eight features, CENT-GO performs better than 23 features of ING-GO with ACC and T evaluation measures, although it does not beat the AUC values (See the result in Table 6.3). Therefore, when all the features are integrated with 31 CENT-ING-GO (combined), the prediction performance is significantly improved. The improvement is confirmed as statistically significant with Mann-Whitney U-statistic test as well.

We compared the prediction performance with different classifier methods as well. The *classifier by Kim et al., 2012*, used in this work, is a meta-classifier using seven decision trees, support vector machine and neural network. For each algorithm, a "bagging" technique is applied to reduce variance and all the eight algorithms are combined by the average rule. The classifier by Kim et al., 2012 in fact improves the performance in the prediction compared with classifier by Acencio and Lemke, 2009, which is a decision tree based meta-classifier used in the study [35].

We analyzed individual features as well to see the impact of each measure compared to all integrated features. When we apply the same classifier to each individual measure, DCGO produces relatively better result than others, although the integration of all eight features perform significantly better. The analysis is conducted by deriving a general rule using a decision tree algorithm as well. We could see that most of decision trees in balanced data sets have MCGO or DCGO as

Table 6.6: Comparison of each measure: The experiment with CENT-GO is compared with the experiment with a single feature each. The performances are compared based on its AUC-ROC, AUC-PR, ACC and T measures.

	AUC-ROC	**AUC-PR**	**ACC**	**T** second
CENT-GO(ALL)	0.784	0.781	0.727	174.19
DCGO	0.760	0.733	0.716	82.53
BCGO	0.711	0.682	0.682	83.34
CCGO	0.742	0.711	0.701	84.82
SCGO	0.732	0.719	0.682	129.89
ECGO	0.718	0.720	0.675	101.64
SoECCGO	0.743	0.741	0.702	115.13
LACGO	0.743	0.738	0.689	101.95
MCGO	0.749	0.747	0.710	96.61

a root node, indicating their important impacts on the general rule. The superiority of MCGO and DCGO has been proven in the previous study [34].

The work has three contributions in the discovery of essential proteins: 1) We combined eight centrality measures computed from PPI network to predict essential proteins using machine learning techniques. Because this feature set includes a smaller number of features than ING-GO (Acencio and Lemke, 2009), the classification process saves computation complexity but produces similar quality of results as ING-GO. 2) We incorporated GO information into the process of centrality measures using EDGEGO, so that additional GO term features are unnecessary. 3) We improved the performance significantly by combining CENT-GO (Kim et al., 2012) and ING-GO (Acencio and Lemke, 2009), with a new design of classifier which adds neural network and support vector machine algorithms.

Prediction of essential proteins can be improved further. First, if we can extract the features from integrated networks, not a single PPI network, then we can obtain enriched feature sets to improve the prediction performance. However, the construction of integrated network requires a large amount of experiments and these networks are available only a limited number of organisms. This limitation hinders the prediction task for various organisms. Second, more robust features can

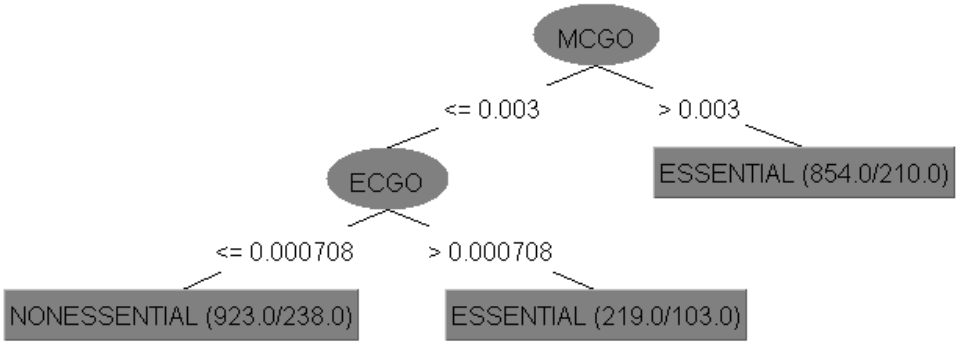


Figure 6.12: Decision tree on the balanced dataset5 with CENT-GO features with 64 instances per leaf: The data set contains only CENT-GO features and the tree algorithm generate a rule where "MCGO" as a root. The values are normalized before running the algorithm, and it produces 72% of accuracy and the area under ROC is $.734$. The eclipses are the features and in this set, "MCGO" and "ECGO" are likely to determine the essentiality of proteins.

improve the prediction task further. Most of centrality measures depend on current GO terms and a PPI network, which are being consistently updated. This limitation can hamper robust experiments and results. Therefore, future studies need to focus on overcoming these limitations.

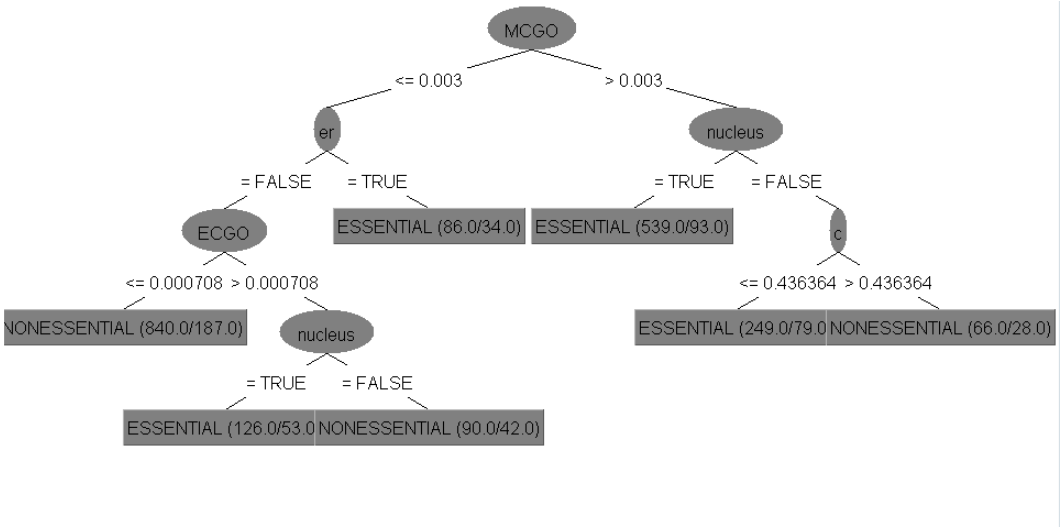


Figure 6.13: Decision tree on the balanced dataset5 with combined features of CENT-ING-GO with 64 instances per leaf: The data set contains 31 features and the tree algorithm generate a rule where "MCGO" as a root. The values are normalized before running the algorithm, and it produces73% of accuracy and the area under ROC is $.752$. The eclipses are the features and in this set, "MCGO" and "ECGO", "clustering coefficient (c)", "nucleus" and "endoplasmic reticulum (er)" are likely to determine the essentiality of proteins.

Chapter 7

CONCLUSIONS

7.1 Conclusions

In this dissertation, we provide new insights for the algorithms and evaluation methods to find biological motifs, including sequence motifs and network motifs. Sequence motifs are substrings in DNA or protein sequences that encode structural motifs or include functional significance. Network motifs are small connected subgraphs in biological networks and they were first introduced as functional building blocks in gene regulatory networks. Sequence motifs have been found through biological or chemical experiments in the past, but now computationally derived motifs are more popular. On the other hand, network motifs are discovered solely through computational methods. Sequence motifs, which are discovered through multiple sequence alignments in a functionally related set of sequences, are 'candidate motifs' until any significant biological function or structural information is verified. Network motifs are defined only by their structural frequency and uniqueness in networks, but there are no comprehensive evaluations to validate biologically important motifs.

We design algorithms which are based on clustering analysis and biological knowledge so that the discovered motifs can be more useful for bioinformatics applications. For protein sequence motifs, we develop an algorithm combining sparse nonnegative matrix factorization (SNMF) method with granular computing and inclusion of statistical structure to discover high quality of protein motifs that are universally conserved across protein family boundaries. They have been applied to the prediction of local tertiary structure [211], for example. Previous algorithms [30–32, 210, 211] have used K-means clustering algorithms and repeated pruning steps for better results based on supervised filtering processes. The initialization process used the secondary structure of the data itself, which should be used only for the evaluation measures. We use an SNMF clustering method for more consistent and efficient results than the previous methods. Additionally, we incorpo-

rated biological knowledge to the data features using Chou-Fasman parameters. To find out their biological roles, we evaluate the candidate motifs with their secondary structure similarity, and additionally suggest a new measurement, sDBI, which evaluates the overall grouping qualities based on the inferred secondary structures and the primary sequences.

For network motifs, we provide new approaches to finding network motifs. Here, we suggest to find biologically meaningful network motifs instead of structural network motifs. As a start, we define a **biological network motif** as a biologically meaningful k-node subgraph. Then we develop efficient algorithms for the detection of biological network motifs and introduce new evaluation measures to assess their biological significance. The algorithms use clustering methods such as Betweenness clustering and SNMF methods. Moreover, some algorithms are biological-knowledge based methods, so that they increase the chance of detecting biological network motifs. All the algorithms introduced in this study improve existing algorithms for high quality of structural network motif detection as well. We also introduce a number of evaluation measures which measure biological significance of each subgraph. We ran the algorithms on two PPI networks of *S.cerevisiae*, and compared the algorithms based on the new measures. An existing exhaustive search and other two existing approximation algorithms are also provided to be compared with our algorithms. As we know of, this is the first time to introduce systematical evaluation measures for network motifs.

We applied the biological network motifs in a practical problem, which is to detect essential proteins in a PPI network. Essential proteins are indispensable to support cellular life and they are a minimal set required for a living cell. They not only help understand the cellular life of an organism, but also are useful for practical usages such as drug design. A number of centrality algorithms have been used to discover essential proteins; degree centrality (DC), betweenness centrality (BC), closeness centrality (CC), subgraph centrality (SC) or eigenvector centrality (EC) However, all the centrality algorithms depend only on the structural properties in a network. In this work, we show that the combination of network motifs and biological annotation improves the detection rates greatly, by proposing a new centrality algorithm, MCGO. We first develop a new centrality algorithm, motif centrality (MC) that counts the number of network motifs for the vertex.

Since network motifs are determined by its statistical significance, MC is more secure algorithm than others to rank vertices in a network. MCGO is MC in an edge-pruned network by EDGEGO, which trims edges based on GO terms. We also provide three evaluation measures to compare the performance with MCGO to those of other centrality algorithms. The evaluation methods include top-ranked true positive rates, statistical measurements and precision-recall curves. We additionally show that the incorporation of gene ontology (GO) annotations improve the performances further with other centrality algorithms.

However, depending only one centrality algorithm for the prediction of essential proteins might be not enough. Rio et al. [292] showed that not one centrality algorithm is dominant in the prediction of essential proteins, but combination of two or more can increase the detection rates. Hence, we make full use of existing centrality measures including MCGO and biological information to predict essential proteins with machine learning techniques in the yeast *Saccharomyces cerevisiae* PPI network. The network is first pruned by EDGEGO algorithm which removes some interactions of relatively uninformative GO terms. From the GO-pruned PPI network, we compute eight centrality measures, namely, DCGO, BCGO, CCGO, ECGO, SCGO, SoECCGO, LACGO and MCGO, where the DC, BC, CC, EC, SC, SoECC, LAC and MC algorithms attach '-GO' term as they are computed from a GO-pruned network. With the eight centrality features, called CENT-GO, we construct ten balanced data sets where the number of essential proteins and the number of non-essential proteins are the same, to avoid biased performance to the majority set. For evaluation measures, we used the area under ROC (AUC-ROC), the area under PR (AUC-PR), accuracy at an optimal threshold (ACC) and the computational time (T). We first confirmed that the 10 balanced data sets are statistically similar through Mann-Whitney U-statistics test, so that we can choose a data set for further experiments. The performance is compared with the data set with 23 features obtained from [35], named, ING-GO (Acencio and Lemke, 2009). With only eight features, CENT-GO (Kim et al. 2012) performs better than 23 features of ING-GO(Acencio and Lemke, 2009) with ACC and T, although it does not beat the AUC values. Therefore, when all the features are integrated, the prediction performance significantly improves on all three evaluation methods.

The improvement is confirmed as statistically significant with Mann-Whitney U-statistic test as well.

We analyzed individual features as well to see the impact of each measure compared to all integrated features. When we apply the same classifier to each individual measure, DCGO produces relatively better results than others, although the integration of all eight features perform significantly better. Another analysis is conducted by deriving a general rule using a decision tree algorithm as well. We could see that most of decision trees in the balanced data sets have MCGO or DCGO as a root node, indicating their important impacts on the general rules. In fact, the good quality of MCGO or DCGO has been proved in Chapter 5.

7.2 Future Work

This research has addressed major issues for biological motif finding and established the following contributions to the study of biological motifs.

- The approach is computation based method. We utilize clustering analysis for the discovery of biological motifs as they are unknown patterns, locations are unknown and the size is unknown. We use the 'intrinsic' similarities in the given data set for more efficient searches.
- We develop biological knowledge-based algorithms to detect biological motifs and introduce a number of evaluation methods to assess their biological qualities.
- We applied our methods for an application of essential protein detection. In a PPI network, integrated feature of network motif and GO term can discover the essential proteins efficiently. Also, the combination of other features improves the prediction performance further using machine learning techniques.

However, other issues still remain to be investigated and studied. Therefore, our future works include the followings. We need to find a way to decide an optimal number of clusters automatically. This is an unsolved major issue in any clustering algorithms and many studies are dedicated to solve the problem. Our work should focus on developing similarity metric of biological data.

Resolving how to assign each data to each cluster when there are one or more candidates is another area of future interest.

The algorithms developed in this research can be improved further. Currently, the parameter-tuning for various algorithms is limited. In the algorithms for biological network motif search, the parameters were adjusted to obtain the desired number of subgraphs to search, for example. In near future, various impacts of the parameters on the results should be investigated. Besides the parameters, the balance between topological and biological information will be an important factor for a better algorithm. On the other hand, current evaluation measures are limited based on the data set, for example, in PPI networks, or in the set of proteins. Comprehensive evaluation measures should be designed to apply various types of biological data. Mostly, this research is based on incomplete biological data. The size of data is still growing, and currently include many false information. We need to design more secure algorithms and evaluation methods.

REFERENCES

[1] F. Harary and E. M. Palmer, *Graphical Enumeration*. Academic Press, 1973, vol. 16, no. 2.

[2] B. McKay, "Nauty user's guide," Dept. of Computer Science, Australian Nat'l Univ., Tech. Rep. TR-CS-90-02, 1990.

[3] A. Krogh, "Chapter 4 an introduction to hidden markov models for biological sequences," in *Biological Sequences, Computational Methods in Molecular Biology, Elsevier*, 1998.

[4] T. D. Schneider and R. M. Stephens, "Sequence logos: A new way to display consensus sequences," *Nucleic Acids Res.*, vol. 18, pp. 6097–6100, 1990.

[5] B. P. Kelley, R. Sharan, R. M. Karp, T. Sittler, D. E. Root, B. R. Stockwell, and T. Ideker, "Conserved pathways within bacteria and yeast as revealed by global protein network alignment," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 20, pp. 11 394–11 399, 2003.

[6] J. Flannick, A. Novak, B. S. Srinivasan, H. H. McAdams, and S. Batzoglou, "Graemlin: General and robust alignment of multiple large interaction networks," *Genome Research*, vol. 16, no. 9, pp. 1169–1181, 2006.

[7] Q. Yang and S.-H. Sze, "Path matching and graph matching in biological networks," *Journal of Computational Biology*, vol. 14, no. 1, pp. 56–67, 2007.

[8] D. Hanahan and R. A. Weinberg, "The hallmarks of cancer," *Cell*, vol. 100, no. 1, pp. 57 – 70, 2000.

[9] E. Klipp and W. Liebermeister, "Mathematical modeling of intracellular signaling pathways," *BMC Neuroscience*, vol. 7, no. Suppl 1, p. S10, 2006.

[10] J. Ruan, Y. Deng, E. Perkins, and W. Zhang, "An ensemble learning approach to reverse-engineering transcriptional regulatory networks from time-series gene expression data," *BMC Genomics*, vol. 10, no. Suppl 1, p. S8, 2009.

[11] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabasi, "The large-scale organization of metabolic networks," *Nature*, vol. 407, pp. 651–654, 2000.

[12] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.

[13] B. H. Junker and F. Schreiber, *Analysis of Biological Networks*. Wiley, 2008.

[14] A. Zanzoni, L. Montecchi-Palazzi, M. Quondam, G. Ausiello, M. Helmer-Citterich, and G. Cesareni, "MINT: a molecular interaction database," *FEBS Letters*, vol. 513, no. 1, pp. 135 – 140, 2002.

[15] S. Wernicke, "Efficient detection of network motifs," *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, vol. 3, no. 4, pp. 347–359, 2006.

[16] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network Motifs: Simple Building Blocks of Complex Networks," *Science*, vol. 298, no. 5594, pp. 824–827, 2002.

[17] L. Parida, "Discovering Topological Motifs Using a Compact Notation," *Journal of Computational Biology*, vol. 14, no. 3, pp. 300–323, 2007.

[18] S. Wernicke and F. Rasche, "FANMOD: a tool for fast network motif detection," *Bioinformatics*, vol. 22, no. 9, pp. 1152–1153, 2006.

[19] J. Chen, W. Hsu, M. L. Lee, and S.-K. Ng, "NeMoFinder: dissecting genome-wide protein-protein interactions with meso-scale network motifs," in *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2006, pp. 106–115.

[20] T. Wang, J. W. Touchman, W. Zhang, E. B. Suh, and G. Xue, "A parallel algorithm for extracting transcription regulatory network motifs," *Bioinformatic and Bioengineering, IEEE International Symposium on*, vol. 0, pp. 193–200, 2005.

[21] M. Schatz, E. Cooper-Balis, and A. Bazinet, "Parallel network motif finding," University of Maryland Insitute for Advanced Computer Studies, Tech. Rep., 2008.

[22] R. Milo, N. Kashtan, S. Itzkovitz, M. E. J. Newman, and U. Alon, "On the uniform generation of random graphs with prescribed degree sequences," 2003.

[23] N. Przulj, D. G. Corneil, and I. Jurisica, "Modeling interactome: scale-free or geometric?" *Bioinformatics*, vol. 20, no. 18, pp. 3508–3515, 2004.

[24] M. Middendorf, E. Ziv, and C. H. Wiggins, "Inferring network mechanisms: The Drosophila melanogaster protein interaction network," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 9, pp. 3192–3197, 2005.

[25] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon, "Superfamilies of Evolved and Designed Networks," *Science*, vol. 303, no. 5663, pp. 1538–1542, 2004.

[26] I. Albert and R. Albert, "Conserved network motifs allow protein-protein interaction prediction." *Bioinformatics*, vol. 20, no. 18, pp. 3346–3352, December 2004.

[27] G. C. Conant and A. Wagner, "Convergent evolution of gene circuits," *Nature Genetics*, vol. 34, pp. 244–266, 2003.

[28] W.-P. Lee, B.-C. Jeng, T.-W. Pai, C.-P. Tsai, C.-Y. Yu, and W.-S. Tzou, "Differential evolutionary conservation of motif modes in the yeast protein interaction network," *BMC Genomics*, vol. 7, no. 1, p. 89, 2006.

[29] J. Chen, W. Hsu, M. L. Lee, and S.-K. Ng, "Labeling network motifs in protein interactomes for protein function prediction," *Data Engineering, International Conference on*, vol. 0, pp. 546–555, 2007.

[30] W. Zhong, G. Altun, R. Harrison, P. Tai, and Y. Pan, "Improved k-means clustering algorithm for exploring local protein sequence motifs representing common structural property," in *IEEE Transactions on Nanobioscience*, vol. 14, no. 3, 2005, pp. 255–265.

[31] B. Chen, P. Tai, R. Harrison, and Y. Pan, "FIK model: A novel efficient granular computing model for protein sequence motifs and structure information discovery," in *The IEEE Symposium on Bioinformatics and Bioengineering*, 2006, pp. 20–26.

[32] B. Chen, P. C. Tai, R. Harrison, and Y. Pan, "FGK model: A efficient granular computing model for protein sequence motifs information discovery," in *The IASTED International Conference on Computational and Systems Biology*, 2006, pp. 56–61.

[33] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon, "Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs," *Bioinformatics*, vol. 20, no. 11, pp. 1746–1758, 2004.

[34] W. Kim, M. Li, J. Wang, and Y. Pan, "Essential protein discovery based on network motif and gene ontology," in *Proceedings of IEEE Bioinformatics and Biomedicine*, 2011, pp. 470–475.

[35] M. Acencio and N. Lemke, "Towards the prediction of essential genes by integration of network topology, cellular localization and biological process information," *BMC Bioinformatics*, vol. 10, no. 1, p. 290, 2009.

[36] Y. Akiyama, T. Hosoya, A. Poole, and Y. Hotta, "The gcm-motif: A novel DNA-binding motif conserved in drosophila and mammals," *Proceedings of the National Academy of Sciences*, vol. 93, no. 25, pp. 14 912–14 916, 1996.

[37] P. D'haeseleer, "What are DNA sequence motifs?" *Nature Biotechnology*, vol. 24, no. 4, pp. 423–425, Apr. 2006.

[38] P. Bork and E. V. Koonin, "Protein sequence motifs," *Current Opinion in Structural Biology*, vol. 6, no. 3, pp. 366 – 376, 1996.

[39] G. D. Stormo, "DNA binding sites: representation and discovery," *Bioinformatics*, vol. 16, no. 1, pp. 16–23, 2000.

[40] A. M. Maxam and W. Gilbert, "A new method for sequencing DNA," *Proceedings of the National Academy of Sciences*, vol. 74, no. 2, pp. 560–564, 1977.

[41] F. Sanger, S. Nicklen, and A. R. Coulson, "DNA sequencing with chain-terminating inhibitors," *Proceedings of the National Academy of Sciences*, vol. 74, no. 12, pp. 5463–5467, 1977.

[42] I. Eidhammer, I. Jonassen, and W. R. Taylor, *Protein Bioinformatics: An Algorithmic Approach to Sequence and Structure Analysis*, 1st ed. Wiley, 2004.

[43] P. Ferreira and P. Azevedo, "Evaluating deterministic motif significance measures in protein databases," *Algorithms for Molecular Biology*, vol. 2, no. 1, pp. 16+, Dec. 2007.

[44] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, May 1998.

[45] N. Hulo, C. Sigrist, L. Saux, P. Langendijk-Genevaux, L. Bordoli, A. Gattiker, E. de Castro, P. Bucher, and A. Bairoch, "Recent improvements to the PROSITE database," *Nucleic Acid Res.*, vol. 32, pp. 134–137, 2004.

[46] S. R. Eddy, "Profile hidden markov models." *Bioinformatics*, vol. 14, no. 9, pp. 755–763, Jan. 1998.

[47] G. Churchill, "Stochastic models for heterogeneous DNA sequences," *Bulletin of Mathematical Biology*, vol. 51, pp. 79–94, 1989.

[48] G. Crooks, G. Hon, J. Chandonia, and S. Brenner, "WebLogo: a sequence logo generator," *Genome Research*, vol. 14, pp. 1188–1190, 2004.

[49] P. D'haeseleer, "How does DNA sequence motif discovery work?" *Nature Biotechnology*, vol. 24, no. 8, pp. 959–961, aug 2006.

[50] G. Pavesi, P. Mereghetti, G. Mauri, and G. Pesole, "Weeder Web: discovery of transcription factor binding sites in a set of sequences from co-regulated genes," *Nucleic Acids Research*, vol. 32, no. suppl 2, pp. W199–W203, 2004.

[51] N. Li and M. Tompa, "Analysis of computational approaches for motif discovery," *Algorithms for Molecular Biology*, vol. 1, no. 1, p. 8, 2006.

[52] P. A. Pevzner and S. H. Sze, "Combinatorial approaches to finding subtle signals in DNA sequences." *Proceedings International Conference on Intelligent Systems for Molecular Biology ISMB International Conference on Intelligent Systems for Molecular Biology*, vol. 8, pp. 269–278, 2000.

[53] G. Pavesi, G. Mauri, and G. Pesole, "An algorithm for finding signals of unknown length in DNA sequences," *Bioinformatics*, vol. 17, no. suppl 1, pp. S207–S214, 2001.

[54] S. Sinha and M. Tompa, "YMF: a program for discovery of novel transcription factor binding sites by statistical overrepresentation," *Nucleic Acids Research*, vol. 31, no. 13, pp. 3586–3588, 2003.

[55] T. L. Bailey and C. Elkan, "Unsupervised learning of multiple motifs in biopolymers using expectation maximization," *Mach. Learn.*, vol. 21, pp. 51–80, October 1995.

[56] ——, "Fitting a mixure model by expectation maximization to discover motifs in biopolymers," in *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, vol. 2. AAAI Press, 1994, pp. 28–36.

[57] W. Ao, J. Gaudet, W. J. Kent, S. Muttumu, and S. E. Mango, "Environmentally induced foregut remodeling by pha-4/foxa and daf-12/nhr," *Science*, vol. 305, no. 5691, pp. 1743–1746, 2004.

[58] C. Lawrence, S. Altschul, M. Boguski, J. Liu, A. Neuwald, and J. Wootton, "Detecting subtle sequence signals: a gibbs sampling strategy for multiple alignment," *Science*, vol. 262, pp. 208–214, 1993.

[59] F. P. Roth, J. D. Hughes, P. W. Estep, and G. M. Church, "Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation." *Nature biotechnology*, vol. 16, no. 10, pp. 939–945, Oct. 1998.

[60] M. C. Frith, U. Hansen, J. L. Spouge, and Z. Weng, "Finding functional sequence elements by multiple local alignment," *Nucleic Acids Research*, vol. 32, no. 1, pp. 189–200, 2004.

[61] G. Thijs, K. Marchal, M. Lescot, S. Rombauts, B. De Moor, P. Rouzé, and Y. Moreau, "A Gibbs sampling method to detect over-represented motifs in the upstream regions of co-expressed genes," in *Proceedings of the fifth annual international conference on Computational biology*, ser. RECOMB '01. New York, NY, USA: ACM, 2001, pp. 305–312.

[62] P. G. Ferreira and P. J. Azevedo, "Evaluating protein motif significance measures: A case study on Prosite patterns," *2007 IEEE Symposium on Computational Intelligence and Data Mining*, no. Cidm, pp. 171–178, 2007.

[63] G. Stolovitzky and A. Califano, "Statistical significance of patterns in biosequences," IBM Computational Biology Center, Tech. Rep., 1998.

[64] A. Brazma, I. Jonassen, I. Eidhammer, and D. Gilbert, "Approaches to the automatic discovery of patterns in biosequences," 1995.

[65] M.-F. Sagot, "On motifs in biological sequences," 2001.

[66] A. Ben-Hur and D. Brutlag, "Sequence motifs: highly predictive features of protein function," in *In Proceedings of Workshop on Feature Selection, Neural Information Processing Systems*, 2003.

[67] T. D. Wu and et al., "Identification of protein motifs using conserved amino acid properties and partitioning techniques," in *Proc. of third international conference on intelligent systems for molecular biology*. AAAI press, 1995, pp. 402–410.

[68] J. Yang, W. Wang, and P. S. Yu, "Mining surprising periodic patterns," *Data Mining and Knowledge Discovery*, vol. 9, pp. 189–216, September 2004.

[69] A. Brazma, I. Jonassen, E. Ukkonen, and J. Vilo, "Discovering patterns and subfamilies in biosequences," in *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, 1996, pp. 34–43.

[70] C. G. Nevill-Manning, K. S. Sethi, T. D. Wu, and D. L. Brutlag, "Enumerating and ranking discrete motifs," in *Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, 1997, pp. 202–209.

[71] P. Smyth and R. Goodman, *Rule Induction Using Information Theory*. MIT press, 1990.

[72] N. M. Abramson, *Information Theory and Coding*. McGraw-Hill, New York, 1963.

[73] G. van den Eijkel, *Intelligent Data Analysis*, M. Certhold and D. J. Hand, Eds. Springer, 2003.

[74] J. D. Hughes, P. W. Estep, S. Tavazoie, and G. M. Church, "Computational identification of cis-regulatory elements associated with groups of functionally related genes in saccharomyces cerevisiae," *Journal of Molecular Biology*, vol. 296, no. 5, pp. 1205–1214, 2000.

[75] A. M. McGuire, J. D. Hughes, and G. M. Church, "Conservation of DNA regulatory motifs and discovery of new motifs in microbial genomes," *Genome Research*, vol. 10, no. 6, pp. 744–757, 2000.

[76] W. Zhong, G. Altun, R. Harrison, P. C. Tai, and Y. Pan, "Improved k-means clustering algorithm for exploring local protein sequence motifs representing common structural property," *NanoBioscience, IEEE Transactions on*, vol. 4, no. 3, pp. 255–265, 2005.

[77] W. Kim, B. Chen, J. Kim, Y. Pan, and H. Park, "Sparse nonnegative matrix factorization for protein sequence motif discovery," *Expert Systems with Applications*, vol. 38, no. 10, pp. 13 198 – 13 207, 2011.

[78] A. Ben-hur and D. Brutlag, "Sequence motifs: Highly predictive features of protein function," *Enzyme*, vol. 645, pp. 625–645, 2006.

[79] A. Ben-Hur and D. Brutlag, "Remote homology detection: a motif based approach," *Bioinformatics*, vol. 19, no. suppl 1, pp. 26–33, 2003.

[80] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Jul. 1999.

[81] G. Bejerano and G. Yona, "Modeling protein families using probabilistic suffix trees," in *Proceedings of the third annual international conference on Computational molecular biology*, ser. RECOMB '99. New York, NY, USA: ACM, 1999, pp. 15–24.

[82] A. Krogh, M. Brown, I. Mian, K. Sjolander, and D. Haussler, "Hidden markov models in computational biology: Applications to protein modeling," *Journal of Molecular Biology*, vol. 235, no. 5, pp. 1501 – 1531, 1994.

[83] P. G. Ferreira and P. J. Azevedo, "Protein sequence classification through relevant sequence mining and bayes classifiers," in *In: Proc. 12th Portuguese Conference on Artificial Intelligence (EPIA*. Springer-Verlag, 2005, pp. 236–247.

[84] K. Blekas, D. I. Fotiadis, and A. Likas, "Motif-based protein sequence classification using neural networks." *Journal of computational biology a journal of computational molecular cell biology*, vol. 12, no. 1, pp. 64–82, 2005.

[85] J. Yang and W. Wang, "CLUSEQ: Efficient and effective sequence clustering," in *In ICDE*. IEEE Press, 2003, pp. 101–112.

[86] S. T. Jensen, L. Shen, and J. S. Liu, "Combining phylogenetic motif discovery and motif clustering to predict co-regulated genes," *Bioinformatics*, vol. 21, pp. 3832–3839, October 2005.

[87] A. P. Heath and L. E. Kavraki, "Computational challenges in systems biology," *Computer Science Review*, vol. 3, no. 1, pp. 1 – 17, 2009.

[88] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon, "Network motifs in the transcriptional regulation network of Escherichia coli," *Nat Genet*, vol. 31, no. 1, pp. 64–68, May 2002.

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

[90] S. Henikoff, J. Henikoff, and S. Pietrokovski, "New features of the blocks database servers," *Nucleic Acid Res*, vol. 27, pp. 226–228, 1999.

[91] ——, "BLOCKS++: a non redundant database of protein alignment blocks derived from multiple compilation," *Bioinformatics*, vol. 15, no. 6, pp. 417–479, 1999.

[92] R. Siddharthan, E. D. Siggia, and E. van Nimwegen, "PhyloGibbs: A Gibbs sampling motif finder that incorporates phylogeny," *PLoS Comput Biol*, vol. 1, no. 7, p. e67, 12 2005.

[93] Q. Zhou and W. H. Wong, "CisModule: De novo discovery of cis-regulatory modules by hierarchical mixture modeling," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 33, pp. 12 114–12 119, 2004.

[94] G. Pavesi, F. Zambelli, and G. Pesole, "WeederH: an algorithm for finding conserved regulatory motifs and regions in homologous sequences," *BMC Bioinformatics*, vol. 8, no. 1, p. 46, 2007.

[95] B. McKay, "Practical graph isomorphism," *Congressus Numerantium*, vol. 30, pp. 45–87, 1981.

[96] J. Grochow and M. Kellis, "Network Motif Discovery Using Subgraph Enumeration and Symmetry-Breaking," *Research in Computational Molecular Biology*, pp. 92–106, 2007.

[97] S. Wuchty, Z. N. Oltvai, and A. Barabasi, "Evolutionary conservation of motif constituents within the yeast protein interaction network," *Nature Genetics*, Oct. 2003.

[98] L. Zhang, O. King, S. Wong, D. Goldberg, A. Tong, G. Lesage, B. Andrews, H. Bussey, C. Boone, and F. Roth, "Motifs, themes and thematic maps of an integrated saccharomyces cerevisiae interaction network," *Journal of Biology*, vol. 4, no. 2, p. 6, 2005.

[99] M. Kuramochi and G. Karypis, "Finding Frequent Patterns in a Large Sparse Graph," *Data Mining and Knowledge Discovery*, vol. 11, no. 3, pp. 243–271, November 2005.

[100] J. Berg and M. Lassig, "Local graph alignment and motif search in biological networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 41, pp. 14 689–14 694, 2004.

[101] S. Mangan and U. Alon, "Structure and function of the feed-forward loop network motif," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 21, pp. 11 980–11 985, 2003.

[102] P. Ingram, M. Stumpf, and J. Stark, "Network motifs: structure does not determine function," *BMC Genomics*, vol. 7, no. 1, p. 108, 2006.

[103] M. Kittisopikul and G. M. Suel, "Biological role of noise encoded in a genetic network motif," *Proceedings of the National Academy of Sciences*, 2010.

[104] N. Bhardwaj and H. Lu, "Co-expression among constituents of a motif in the protein-protein interaction network," *J. Bioinformatics and Computational Biology*, vol. 7, no. 1, pp. 1–17, 2009.

[105] R. J. Prill, P. A. Iglesias, and A. Levchenko, "Dynamic properties of network motifs contribute to biological network organization," *PLoS Biol*, vol. 3, no. 11, p. e343, 10 2005.

[106] J. Hallinan and A. Wipat, "Network motifs in context: An exploration of the evolution of oscillatory dynamics in transcriptional networks," in *Computational Intelligence in Bioinformatics and Computational Biology*, 2008, p. 83.

[107] Y. Zhang, J. Xuan, B. G. de los Reyes, R. Clarke, and H. W. Ressom, "Network motif-based identification of breast cancer susceptibility genes," in *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, Aug. 2008, pp. 5696–5699.

[108] R. Dobrin, Q. K. Beg, A.-L. Barabasi, and Z. N. Oltvai, "Aggregation of topological motifs in the Escherichia coli transcriptional regulatory network," *BMC Bioinformatics*, vol. 5, p. 10, 2004.

[109] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon, "Topological generalizations of network motifs," *Physical Review E*, vol. 70, p. 031909, 2004.

[110] Z.-R. Xie and M.-J. Hwang, "An interaction-motif-based scoring function for protein-ligand docking," *BMC Bioinformatics*, vol. 11, no. 1, p. 298, 2010.

[111] Y. Artzy-Randrup, S. J. Fleishman, N. Ben-Tal, and L. Stone, "Comment on "network motifs: Simple building blocks of complex networks" and "superfamilies of evolved and designed networks"," *Science*, vol. 305, no. 5687, p. 1107, 2004.

[112] W.-P. Lee and W.-S. Tzou, "Fast revelation of the motif mode for a yeast protein interaction network through intelligent agent-based distributed computing," *Protein and Peptide Letters*, vol. 17, pp. 1091–1101(11), 2010.

[113] B. Andreopoulos, A. An, X. Wang, and M. Schroeder, "A roadmap of clustering algorithms: finding a match for a biomedical application," *Briefings in Bioinformatics*, vol. 10, no. 3, pp. 297–314, 2009.

[114] A. Vazquez, R. Dobrin, D. Sergi, J.-P. Eckmann, Z. N. Oltvai, and A.-L. Barabasi, "The topological relationship between the large-scale attributes and local interaction patterns of complex networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 52, pp. 17 940–17 945, 2004.

[115] M. Habibi, C. Eslahchi, and L. Wong, "Protein complex prediction based on k-connected subgraphs in protein interaction network," *BMC Systems Biology*, vol. 4, no. 1, p. 129, 2010.

[116] J.Wang and G. Provan, "On motifs and functional modules in complex networks," in *Science and Technology for Humanity (TIC-STH), 2009 IEEE Toronto International Conference*, 2009, pp. 78–82.

[117] A. M. Lesk, *Introduction to Bioinformatics*. Oxford University Press, May 2002.

[118] H. Kitano, *Foundations of Systems Biology*. Cambridge, MA.: The MIT Press, 2001.

[119] D. Noble, *The Music of Life: Biology Beyond Genes*. Oxford University Press, USA, Jul. 2006.

[120] L. Chong and L. B. Ray, "Whole-istic biology," *Science*, vol. 295, no. 5560, p. 1661, 2002.

[121] "-omes and -omics glossaru and taxamony[$http://www.genomicglossaries.com$]."

[122] H. Kitano, "Systems biology: A brief overview," *Science*, vol. 295, no. 5560, pp. 1662–1664, 2002.

[123] E. Davidson and M. Levin, "Gene regulatory networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 14, p. 4935, 2005.

[124] M. Kanehisa, M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, T. Tokimatsu, and Y. Yamanishi, "KEGG for linking genomes to life and the environment," *Nucleic Acids Research*, vol. 36, no. suppl 1, pp. D480–D484, 2008.

[125] I. M. Keseler, J. Collado-Vides, S. Gama-Castro, J. Ingraham, S. Paley, I. T. Paulsen, M. Peralta-Gil, and P. D. Karp, "EcoCyc: a comprehensive database resource for escherichia coli," *Nucleic Acids Research*, vol. 33, no. suppl 1, pp. D334–D337, 2005.

[126] E. A. Ananko, N. L. Podkolodny, I. L. Stepanenko, O. A. Podkolodnaya, D. A. Rasskazov, D. S. Miginsky, V. A. Likhoshvai, A. V. Ratushny, N. N. Podkolodnaya, and N. A. Kolchanov, "GeneNet in 2005," *Nucleic Acids Research*, vol. 33, no. suppl 1, pp. D425–D427, 2005.

[127] J. Segura-salazar, L. Muniz-rascado, I. Martinez-flores, H. Salgado, C. Bonavides-martinez, C. Abreu-goodger, C. Rodriguez-penagos, J. Mir, E. Morett, E. Merino, A. M. Huerta,

L. Trevino-quintanilla, and J. Collado-vides, "RegulonDB (version 6.0): gene regulation model," 2007.

[128] V. Matys, E. Fricke, R. Geffers, E. Goling, M. Haubrock, R. Hehl, K. Hornischer, D. Karas, A. E. Kel, O. V. Kel-Margoulis, D.-U. Kloos, S. Land, B. Lewicki-Potapov, H. Michael, R. Munch, I. Reuter, S. Rotert, H. Saxel, M. Scheer, S. Thiele, and E. Wingender, "TRANSFAC: transcriptional regulation, from patterns to profiles," *Nucleic Acids Research*, vol. 31, no. 1, pp. 374–378, 2003.

[129] D. U. Silverthorn, *Human Physiology: An Integrated Approach*. Pearson Higher Education, 2006.

[130] M. Beato, S. Chávez, and M. Truss, "Transcriptional regulation by steroid hormones," *Steroids*, vol. 61, no. 4, pp. 240 – 251, 1996.

[131] C. F. Schaefer, K. Anthony, S. Krupa, J. Buchoff, M. Day, T. Hannay, and K. H. Buetow, "PID: the pathway interaction database." *Nucleic acids research*, vol. 37, no. Database issue, pp. D674–D679, 2009.

[132] R. Elkon, R. Vesterman, N. Amit, I. Ulitsky, I. Zohar, M. Weisz, G. Mass, N. Orlev, G. Sternberg, R. Blekhman, J. Assa, Y. Shiloh, and R. Shamir, "SPIKE - a database, visualization and analysis tool of cellular signaling pathways," *BMC Bioinformatics*, vol. 9, no. 1, p. 110, 2008.

[133] M. Krull, S. Pistor, N. Voss, A. Kel, I. Reuter, D. Kronenberg, H. Michael, K. Schwarzer, A. Potapov, C. Choi, O. Kel-Margoulis, and E. Wingender, "TRANSPATH: an information resource for storing and visualizing signaling pathways and their pathological aberrations." *Nucleic acids research*, vol. 34, no. Database issue, Jan. 2006.

[134] M. Krull, N. Voss, C. Choi, S. Pistor, A. Potapov, and E. Wingender, "TRANSPATH: an integrated database on signal transduction and a tool for array analysis," *Nucleic Acids Research*, vol. 31, no. 1, pp. 97–100, 2003.

[135] V. Matys, O. V. Kel-Margoulis, E. Fricke, I. Liebich, S. Land, A. Barre-Dirrie, I. Reuter, D. Chekmenev, M. Krull, K. Hornischer, N. Voss, P. Stegmaier, B. Lewicki-Potapov, H. Saxel, A. E. Kel, and E. Wingender, "TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes," *Nucleic Acids Research*, vol. 34, no. suppl 1, pp. D108–D110, 2006.

[136] T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J.-B. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford, and R. A. Young, "Transcriptional Regulatory Networks in Saccharomyces cerevisiae," *Science*, vol. 298, no. 5594, pp. 799–804, 2002.

[137] C. Francke, R. J. Siezen, and B. Teusink, "Reconstructing the metabolic network of a bacterium from its genome," *Trends in Microbiology*, vol. 13, no. 11, pp. 550 – 558, 2005.

[138] R. Caspi, H. Foerster, C. A. Fulcher, P. Kaipa, M. Krummenacker, M. Latendresse, S. Paley, S. Y. Rhee, A. G. Shearer, C. Tissier, T. C. Walk, P. Zhang, and P. D. Karp, "The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases," *Nucleic Acids Research*, vol. 36, no. suppl 1, pp. D623–D631, 2008.

[139] P. D. Karp, C. A. Ouzounis, C. Moore-Kochlacs, L. Goldovsky, P. Kaipa, D. Ahren, S. Tsoka, N. Darzentas, V. Kunin, and N. Lopez-Bigas, "Expansion of the biocyc collection of pathway/genome databases to 160 genomes," *Nucleic Acids Research*, vol. 33, no. 19, pp. 6083–6089, 2005.

[140] H. Ge, A. Walhout, and M. Vidal, "Integrating 'omic' information: a bridge between genomics and systems biology." *Trends in genetics : TIG*, vol. 19, no. 10, pp. 551–560, Oct. 2003.

[141] E. Phizicky and S. Fields, "Protein-protein interactions: methods for detection and analysis," *Microbiol. Rev.*, vol. 59, no. 1, pp. 94–123, 1995.

[142] S. Fields and O. kyu Song, "A novel genetic system to detect protein-protein interactions," *Nature*, vol. 340, pp. 245–246, 1989.

[143] L. Skrabanek, H. Saini, G. Bader, and A. Enright, "Computational prediction of protein-protein interactions," *Molecular Biotechnology*, vol. 38, pp. 1–17, 2008.

[144] M. W. Hahn and A. D. Kern, "Comparative genomics of centrality and essentiality in three eukaryotic protein-interaction networks," *Molecular Biology and Evolution*, vol. 22, no. 4, pp. 803–806, 2005.

[145] T. B. Fischer, M. Paczkowski, M. F. Zettel, and J. Tsai, "A guide to protein interaction databases," in *The Proteomics Protocols Handbook*, J. M. Walker, Ed. Humana Press, 2005, pp. 753–799.

[146] I. Xenarios, D. W. Rice, L. Salwinski, M. K. Baron, E. M. Marcotte, and D. Eisenberg, "DIP: the database of interacting proteins," *Nucleic Acids Research*, vol. 28, no. 1, pp. 289–291, 2000.

[147] G. D. Bader, I. Donaldson, C. Wolting, B. F. F. Ouellette, T. Pawson, and C. W. V. Hogue, "BIND : The biomolecular interaction network database," *Nucleic Acids Research*, vol. 29, no. 1, pp. 242–245, 2001.

[148] Z. L. Ji, X. Chen, C. J. Zhen, L. X. Y. L. Y. Han, W. K. Yeo, P. C. Chung, H. S. Puy, Y. T. Tay, A. Muhammad, and Y. Z. Chen, "KDBI: Kinetic data of bio-molecular interactions database," *Nucleic Acids Res*, vol. 31, no. 1, pp. 255–257, 2003.

[149] T. Liu, Y. Lin, X. Wen, R. N. Jorissen, and M. K. Gilson, "BindingDB: a web-accessible database of experimentally determined protein-ligand binding affinities," *Nucleic Acids Research*, vol. 35, no. suppl 1, pp. D198–D201, 2006.

[150] H. W. Mewes, D. Frishman, U. Guldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenstern, M. Munsterkotter, S. Rudd, and B. Weil, "MIPS: a database for genomes and protein sequences," *Nucleic Acids Research*, vol. 30, no. 1, pp. 31–34, 2002.

[151] S. Peri, J. D. Navarro, R. Amanchy, T. Z. Kristiansen, C. K. Jonnalagadda, V. Surendranath, V. Niranjan, B. Muthusamy, T. Gandhi, M. Gronborg, N. Ibarrola, N. Deshpande, K. Shanker, H. Shivashankar, B. Rashmi, M. Ramya, Z. Zhao, K. Chandrika, N. Padma, H. Harsha, A. Yatish, M. Kavitha, M. Menezes, D. R. Choudhury, S. Suresh, N. Ghosh, R. Saravana, S. Chandran, S. Krishna, M. Joy, S. K. Anand, V. Madavan, A. Joseph, G. W. Wong, W. P. Schiemann, S. N. Constantinescu, L. Huang, R. Khosravi-Far, H. Steen, M. Tewari, S. Ghaffari, G. C. Blobe, C. V. Dang, J. G. Garcia, J. Pevsner, O. N. Jensen, P. Roepstorff, K. S. Deshpande, A. M. Chinnaiyan, A. Hamosh, A. Chakravarti, and A. Pandey, "Development of human protein reference database as an initial platform for approaching systems biology in humans," *Genome Research*, vol. 13, no. 10, pp. 2363–2371, 2003.

[152] K. S. Thorn and A. A. Bogan, "ASEdb: a database of alanine mutations and their effects on the free energy of binding in protein interactions," *Bioinformatics*, vol. 17, no. 3, pp. 284–285, 2001.

[153] T. B. Fischer, K. V. Arunachalam, D. Bailey, V. Mangual, S. Bakhru, R. Russo, D. Huang, M. Paczkowski, V. Lalchandani, C. Ramachandra, B. Ellison, S. Galer, J. Shapley, E. Fuentes, and J. Tsai, "The binding interface database (bid): a compilation of amino acid hot spots in protein interfaces," *Bioinformatics*, vol. 19, no. 11, pp. 1453–1454, 2003.

[154] B.-J. Breitkreutz, C. Stark, and M. Tyers, "The GRID: The general repository for interaction datasets," *Genome Biology*, vol. 4, no. 3, p. R23, 2003.

[155] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabasi, "The large-scale organization of metabolic networks," 2000.

[156] A.-L. Barabasi and R. Albert, "Emergence of Scaling in Random Networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[157] S. Milgram, "The Small World Problem," *Psychology Today*, vol. 2, pp. 60–67, 1967.

[158] R. Z. Albert and A. laszlo Barabasi Director, "Statistical mechanics of complex networks," 2001.

[159] V. Lacroix, L. Cottret, P. Thebault, and M.-F. Sagot, "An introduction to metabolic networks and their structural analysis." *IEEE/ACM Trans. Comput. Biology Bioinform.*, pp. 594–617, 2008.

[160] M. Arita, "The metabolic world of escherichia coli is not small," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 6, pp. 1543–1547, 2004.

[161] ——, "Scale-freeness and biological networks," *Journal of Biochemistry*, vol. 138, no. 1, pp. 1–4, Jul. 2005.

[162] X. He and J. Zhang, "Why do hubs tend to be essential in protein networks?" *PLoS Genet*, vol. 2, no. 6, p. e88, 06 2006.

[163] P. M. Kim, L. J. Lu, Y. Xia, and M. B. Gerstein, "Relating three-dimensional structures to protein networks provides evolutionary insights," *Science*, vol. 314, no. 5807, pp. 1938–1941, 2006.

[164] C. C. Fowlkes, C. L. L. Hendriks, S. V. E. Keranen, G. H. Weber, O. Rubel, M. yu Huang, S. Chatoor, A. H. Depace, L. Simirenko, A. Beaton, R. Weiszmann, S. Celniker, B. Hamann, D. W, M. D. Biggin, M. B. Eisen, and J. Malik, "A quantitative spatiotemporal atlas of gene expression in the drosophila blastoderm," *Cell*, vol. 133, no. 2, pp. 364 – 374, 2008.

[165] J.-D. J. Han, N. Bertin, T. Hao, D. S. Goldberg, G. F. Berriz, L. V. Zhang, D. Dupuy, A. J. M. Walhout, M. E. Cusick, F. P. Roth, and et al., "Evidence for dynamically organized modularity in the yeast protein-protein interaction network." *Nature*, vol. 430, no. 6995, pp. 88–93, 2004.

[166] B. C. Goodwin and S. A. Kauffman, "Spatial harmonics and pattern specification in early drosophila development. part i. bifurcation sequences and gene expression," *Journal of Theoretical Biology*, vol. 144, no. 3, pp. 303 – 319, 1990.

[167] C. L. Myers and O. G. Troyanskaya, "Context-sensitive data integration and prediction of biological networks," *Bioinformatics*, vol. 23, no. 17, pp. 2322–2330, 2007.

[168] J. Rachlin, D. D. Cohen, C. Cantor, and S. Kasif, "Biological context networks: a mosaic view of the interactome," *Mol Syst Biol*, vol. 2, no. 66, 2006.

[169] V. Spirin, M. S. Gelfand, A. A. Mironov, and L. A. Mirny, "A metabolic network in the evolutionary context: Multiscale structure and modularity," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8774–8779, 2006.

[170] T. Knijnenburg, L. Wessels, and M. Reinders, "Combinatorial influence of environmental parameters on transcription factor activity," *Bioinformatics*, vol. 24, no. 13, pp. i172–i181, 2008.

[171] D. Hwang, A. G. Rust, S. Ramsey, J. J. Smith, D. M. Leslie, A. D. Weston, P. de Atauri, J. D. Aitchison, L. Hood, A. F. Siegel, and H. Bolouri, "A data integration methodology for systems biology," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 48, pp. 17 296–17 301, 2005.

[172] D. Hwang, J. J. Smith, D. M. Leslie, A. D. Weston, A. G. Rust, S. Ramsey, P. de Atauri, A. F. Siegel, H. Bolouri, J. D. Aitchison, and L. Hood, "A data integration methodology for systems biology: Experimental verification," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 48, pp. 17 302–17 307, 2005.

[173] T. Y. T. Y. Lin, "Data mining and machine oriented modeling: A granular computing approach," *Applied Intelligence*, vol. 13, no. 2, pp. 113–124, 2000.

[174] Y. Yao, "On modeling data mining with granular computing," in *COMPAC*, 2001, pp. 638–643.

[175] B. S. Everitt, *Cluster analysis*. Heinemann Educational [for] the Social Science Research Council, London, 1974.

[176] R. Xu and D. W. II, "Survey of clustering algorithms," *Neural Networks, IEEE Transactions on*, vol. 16, no. 3, pp. 645–678, May 2005.

[177] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[178] D. Hochbaum and D. Shmoys, "A best possible heuristic for the k-center problem," *Math. Operat. Res.*, vol. 10, pp. 180–184, 1985.

[179] L. Kaufmann and P. Rousseeuw, "Clustering by means of medoids," *Elsevier Science*, pp. 405–416, 1987.

[180] ——, *Finding Groups in Data: An introduction to Cluster Analysis*. John Wiley, 1990.

[181] R. T. Ng and J. Han, "Efficient and effective clustering methods for spatial data mining," in *Proceedings of the 20th International Conference on Very Large Data Bases*, ser. VLDB '94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994, pp. 144–155.

[182] F. Höpner, F. Klawonn, and R. Kruse, *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis, and Image Recognittion*. New York, NY.: Wiley, 1999.

[183] Z. Huang, "A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining," in *In Research Issues on Data Mining and Knowledge Discovery*, 1997, pp. 1–8.

[184] Z. Huang and M. Ng, "A fuzzy k-modes algorithm for clustering categorical data." in *IEEE Trans Fuzzy Syst*, 1999, pp. 446–452.

[185] M. Anderberg, *Cluster Analysis for Applications*. New York, NY: Academic Press, Inc., 1973.

[186] H. Kim and H. Park, "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis," *Bioinformatics*, vol. 23, no. 12, pp. 1495–1502, 2007.

[187] M. L. Zhang, M. W. Edu, T. Zhang, T. Zhang, R. Ramakrishnan, R. Ramakrishnan, and M. Livny, "BIRCH: A new data clustering algorithm and its applications," *Data Mining and Knowledge Discovery*, vol. 1, pp. 141–182, 1997.

[188] S. Guha, R. Rastogi, and K. Shim, "CURE: an efficient clustering algorithm for large databases," *Information Systems*, vol. 26, no. 1, pp. 35 – 58, 2001.

[189] S. White and P. Smyth, "A spectral clustering approach to finding communities in graphs," in *Proceedings of the SDM '05*, 2005.

[190] S. G. Rajeev, R. Rastogi, and K. Shim, "ROCK: A robust clustering algorithm for categorical attributes," *Information system*, vol. 25, pp. 345–366, 2000.

[191] P. Andritsos, P. Tsaparas, and R. Miller, "LIMBO: Scalable clustering of categorical data," in *Proceedings of the 9th international conference on Extending Database Technology*, 2004.

[192] J. Holland, *Adaption in Natural and Artificial Systems*. University of Michigan Press, 1975.

[193] E. R. Hansen, "Numerical optimization of computer models (hans-paul schwefel)," *SIAM Review*, vol. 25, no. 3, pp. 431–433, 1983.

[194] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. of 2nd International Conference on Knowledge Discovery and*. AAAI Press, 1996, pp. 226–231.

[195] R. Agrawal, J. Gehrke, and D. Gunopulos, "Automatic subspace clustering of high dimensional data for data mining applications," in *Proceedings of the ACM-SIGMOD'98 International Conference on Management of Data*, Seattle, WA, 1998, pp. 94–105.

[196] B. Andreopoulos, A. An, and X. Wang, "Hierarchical density-based clustering of categorical data and a simplification," in *Proceedings of the 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Nanjing, China: Springer LNCS, 2007, pp. 11–22.

[197] P. Cheeseman and J. Stutz, *Bayesian Classification (AutoClass): Theory and Results*. AAAI Press/MIT Press, 1996, ch. 6, pp. 62–83.

[198] B. H. Junker and F. Schreiber, *Analysis of Biological Networks*. Wiley, 2008.

[199] D. Jiang, C. Tang, and A. Zhang, "Cluster analysis for gene expression data: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, pp. 1370–1386, 2004.

[200] E. J. Chesler and M. A. Langston, "Combinatorial genetic regulatory network analysis tools for high throughput transcriptomic data," in *Proceedings of the 2005 joint annual satellite conference on Systems biology and regulatory genomics*, ser. RECOMB'05. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 150–165.

[201] G. Kochenberger, F. Glover, B. Alidaee, and H. Wang, "Clustering of microarray data via clique partitioning," *Journal of Combinatorial Optimization*, vol. 10, pp. 77–92, 2005.

[202] X. Peng, M. A. Langston, A. M. Saxton, N. E. Baldwin, and J. R. Snoddy, "Detecting network motifs in gene co-expression networks," 2004.

[203] B. Balasundaram and S. Butenko, "Graph domination, coloring and cliques in telecommunications," in *Handbook of Optimization in Telecommunications*, M. G. C. Resende and P. M. Pardalos, Eds. Springer US, 2006, pp. 865–890.

[204] Y. P. Chen, A. L. Liestman, and J. Liu, "Clustering algorithms for ad hoc wireless networks," in *Ad Hoc and Sensor Networks. Nova Science Publishers*, 2004.

[205] S. Butenko, X. Cheng, C. A. S. Oliveira, and P. M. Pardalos, "A new heuristic for the minimum connected dominating set problem on ad hoc wireless networks," in *Recent Developments in Cooperative Control and Optimization*. Kluwer Academic Publishers,, 2004, pp. 61–73.

[206] D. S. Hochbaum and D. B. Shmoys, "A Best Possible Heuristic for the k-Center Problem," *MATHEMATICS OF OPERATIONS RESEARCH*, vol. 10, no. 2, pp. 180–184, May 1985.

[207] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *nat*, vol. 435, pp. 814–818, jun 2005.

[208] B. Adamcsek, G. Palla, I. J. Farkas, I. Derenyi, and T. Vicsek, "CFinder: locating cliques and overlapping modules in biological networks," *Bioinformatics*, vol. 22, no. 8, pp. 1021–1023, 2006.

[209] C. Zhang, S. Liu, and Y. Zhou, "Fast and accurate method for identifying high-quality protein-interaction modules by clique merging and its application to yeast," *Journal of Proteome Research*, vol. 5, no. 4, pp. 801–807, 2006.

[210] B. Chen, S. Pellicer, P. C. Tai, R. Harrison, and Y. 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*, pp. 8–25, 2008.

[211] B. Chen and M. Johnson, "Protein local 3D structure prediction by super granule support vector machines (Super GSVM)," *BMC Bioinformatics*, vol. 10, no. Suppl 11, p. S15, 2009.

[212] K. Han and D. Baker, "Recurring local sequence motifs in proteins," *Molecular Biology*, vol. 251, pp. 2577–2637, 1983.

[213] H. Kim and H. Park, "Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 2, pp. 713–730, 2008.

[214] K. Devarajan, "Nonnegative matrix factorization: An analytical and interpretive tool in computational biology," *PLoS Comput Biol*, vol. 4, no. 7, p. e1000029, Jul 2008.

[215] J. Kim and H. Park, "Sparse nonnegative matrix factorization for clustering," Computational Science and Engineering, Georgia Institute of Technology, Tech. Rep. GT-CSE-08-01, 2008.

[216] J. M. Peña, J. A. Lozano, and P. Larrañaga, "An empirical comparison of four initialization methods for the k-means algorithm," *Pattern Recogn. Lett.*, vol. 20, no. 10, pp. 1027–1040, 1999.

[217] E. W. Forgy, "Cluster analysis of multivariate data: efficiency vs interpretability of classifications," *Biometrics*, vol. 21, pp. 768–769, 1965.

[218] J. B. Macqueen, "Some methods of classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

[219] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis (Wiley Series in Probability and Statistics)*. Wiley-Interscience, March 2005.

[220] J. Dunn, "A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters," *Cybernetics*, vol. 3, pp. 32–57, 1973.

[221] J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press, 1981.

[222] P. Y. Chou and G. D. Fasman, "Prediction of protein conformation," *Biochemistry*, vol. 13, no. 2, pp. 222–245, 1974.

[223] ——, "Prediction of the secondary structure of proteins from their amino acid sequence," *Adv Enzyol Relat Areas Mol. Biol.*, vol. 47, pp. 45–148, 1978.

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

[225] D. Davies and D. Bouldin, "A cluster separation measure," in *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 1, 1979, pp. 224–227.

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

[227] W. Kabsh and C. Sander, "Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features," *Bioploymers*, vol. 22, pp. 2577–2637, 1979.

[228] A.-L. Barabasi and Z. N. Oltvai, "Network biology: understanding the cell's functional organization," *Nat Rev Genet*, vol. 5, no. 2, pp. 101–113, February 2004.

[229] S. Mangan, A. Zaslaver, and U. Alon, "The coherent feedforward loop serves as a sign-sensitive delay element in transcription networks," *Journal of Molecular Biology*, vol. 334, no. 2, pp. 197 – 204, 2003.

[230] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, 2002.

[231] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium." *Nature genetics*, vol. 25, no. 1, pp. 25–29, May 2000.

[232] J. Wang, M. Li, Y. Deng, and Y. Pan, "Recent advances in clustering methods for protein interaction networks," *BMC Genomics*, vol. 11, no. Suppl 3, p. S10, 2010.

[233] F. Wu and B. A. Huberman, "Finding communities in linear time: a physics approach," *The European Physical Journal B - Condensed Matter and Complex Systems*, vol. 38, no. 2, pp. 331–338, March 2004.

[234] G. Kirchhoff, K. Hensel, and M. Planck, *Vorlesungen uber mathematische Physik*. B.G. Teubner, 1894.

[235] Y. Zhang, E. Zeng, T. Li, and G. Narasimhan, "Weighted consensus clustering for identifying functional modules in protein-protein interaction networks," in *Proceedings of the 2009 International Conference on Machine Learning and Applications*, ser. ICMLA '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 539–544.

[236] C. von Mering, R. Krause, B. Snel, M. Cornell, S. G. Oliver, S. Fields, and P. Bork, "Comparative assessment of large-scale data sets of protei-protein interactions," *Nature*, vol. 417, no. 6887, pp. 399–403, May 2002.

[237] J. Wang, M. Li, J. Chen, and Y. Pan, "A fast hierarchical clustering algorithm for functional modules discovery in protein interaction networks," *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, vol. 8, no. 3, pp. 607 –620, may-june 2011.

[238] K. Kobayashi, S. D. Ehrlich, A. Albertini, G. Amati, K. K. Andersen, M. Arnaud, K. Asai, S. Ashikaga, S. Aymerich, P. Bessieres, F. Boland, S. C. Brignell, S. Bron, K. Bunai, J. Chapuis, L. C. Christiansen, A. Danchin, M. Debarbouille, E. Dervyn, E. Deuerling, K. Devine, S. K. Devine, O. Dreesen, J. Errington, S. Fillinger, S. J. Foster, Y. Fujita, A. Galizzi, R. Gardan, C. Eschevins, T. Fukushima, K. Haga, C. R. Harwood, M. Hecker, D. Hosoya, M. F. Hullo, H. Kakeshita, D. Karamata, Y. Kasahara, F. Kawamura, K. Koga, P. Koski, R. Kuwana, D. Imamura, M. Ishimaru, S. Ishikawa, I. Ishio, D. Le Coq, A. Masson, C. Mauel, R. Meima, R. P. Mellado, A. Moir, S. Moriya, E. Nagakawa, H. Nanamiya, S. Nakai, P. Nygaard, M. Ogura, T. Ohanan, M. O'Reilly, M. O'Rourke, Z. Pragai, H. M. Pooley, G. Rapoport, J. P. Rawlins, L. A. Rivas, C. Rivolta, A. Sadaie, Y. Sadaie, M. Sarvas, T. Sato, H. H. Saxild, E. Scanlan, W. Schumann, J. F. M. L. Seegers, J. Sekiguchi, A. Sekowska, S. J. Seror, M. Simon, P. Stragier, R. Studer, H. Takamatsu, T. Tanaka, M. Takeuchi, H. B. Thomaides, V. Vagner, J. M. van Dijl, K. Watabe, A. Wipat, H. Yamamoto, M. Yamamoto, Y. Yamamoto, K. Yamane, K. Yata, K. Yoshida, H. Yoshikawa, U. Zuber, and N. Ogasawara, "Essential bacillus subtilis genes," *Proceedings of the National Academy of Sciences*, vol. 100, no. 8, pp. 4678–4683, 2003.

[239] M. Itaya, "An estimation of minimal genome size required for life," *FEBS Letters*, vol. 362, no. 3, pp. 257 – 260, 1995.

[240] E. A. Winzeler, D. D. Shoemaker, A. Astromoff, H. Liang, K. Anderson, B. Andre, R. Bangham, R. Benito, J. D. Boeke, H. Bussey, A. M. Chu, C. Connelly, K. Davis, F. Dietrich, S. W. Dow, M. El Bakkoury, F. Foury, S. H. Friend, E. Gentalen, G. Giaever, J. H. Hegemann, T. Jones, M. Laub, H. Liao, N. Liebundguth, D. J. Lockhart, A. Lucau-Danila, M. Lussier, N. M'Rabet, P. Menard, M. Mittmann, C. Pai, C. Rebischung, J. L. Revuelta, L. Riles, C. J. Roberts, P. Ross-MacDonald, B. Scherens, M. Snyder, S. Sookhai-Mahadeo, R. K. Storms, S. Veronneau, M. Voet, G. Volckaert, T. R. Ward, R. Wysocki, G. S. Yen, K. Yu, K. Zimmermann, P. Philippsen, M. Johnston, and R. W. Davis, "Functional characterization of the s. cerevisiae genome by gene deletion and parallel analysis," *Science*, vol. 285, no. 5429, pp. 901–906, 1999.

[241] N. Judson and J. Mekalanos, "TnAraOut, a transposon-based approach to identify and characterize essential bacterial genes." *Nat Biotechnol*, vol. 18, no. 7, pp. 740–5, 2000.

[242] K. Kemphues, "Essential genes," *WormBook*, 2005.

[243] G. Giaever, A. M. Chu, L. Ni, C. Connelly, L. Riles, S. Véronneau, S. Dow, A. Lucau-Danila, K. Anderson, B. André, A. P. Arkin, A. Astromoff, M. El-Bakkoury, R. Bangham, R. Benito, S. Brachat, S. Campanaro, M. Curtiss, K. Davis, A. Deutschbauer, K.-D. D. Entian, P. Flaherty, F. Foury, D. J. Garfinkel, M. Gerstein, D. Gotte, U. Güldener, J. H. Hegemann, S. Hempel, Z. Herman, D. F. Jaramillo, D. E. Kelly, S. L. Kelly, P. Kötter, D. LaBonte, D. C. Lamb, N. Lan, H. Liang, H. Liao, L. Liu, C. Luo, M. Lussier, R. Mao, P. Menard, S. L. L. Ooi, J. L. Revuelta, C. J. Roberts, M. Rose, P. Ross-Macdonald, B. Scherens, G. Schimmack, B. Shafer, D. D. Shoemaker, S. Sookhai-Mahadeo, R. K. Storms, J. N. Strathern, G. Valle, M. Voet, G. Volckaert, C.-y. Y. Wang, T. R. Ward, J. Wilhelmy, E. A. Winzeler, Y. Yang, G. Yen, E. Youngman, K. Yu, H. Bussey, J. D. Boeke, M. Snyder, P. Philippsen, R. W. Davis, and M. Johnston, "Functional profiling of the Saccharomyces cerevisiae genome." *Nature*, vol. 418, no. 6896, pp. 387–391, Jul. 2002.

[244] L. M. Cullen and G. M. Arndt, "Genome-wide screening for gene function using RNAi in mammalian cells," *Immunology and Cell Biology*, vol. 83, no. 3, pp. 217–223, Jun. 2005.

[245] T. Roemer, B. Jiang, J. Davison, T. Ketela, K. Veillette, A. Breton, F. Tandia, A. Linteau, S. Sillaots, C. Marta, N. Martel, S. Veronneau, S. Lemieux, S. Kauffman, J. Becker, R. Storms, C. Boone, and H. Bussey, "Large-scale essential gene identification in candida albicans and applications to antifungal drug discovery," *Molecular Microbiology*, vol. 50, no. 1, pp. 167–181, 2003.

[246] R. Zhang, H.-Y. Ou, and C.-T. Zhang, "DEG: a database of essential genes," *Nucleic Acids Research*, vol. 32, no. suppl 1, pp. D271–D272, 2004.

[247] J. M. Cherry, C. Adler, C. Ball, S. A. Chervitz, S. S. Dwight, E. T. Hester, Y. Jia, G. Juvik, T. Roe, M. Schroeder, S. Weng, and D. Botstein, "SGD: Saccharomyces genome database," *Nucleic Acids Research*, vol. 26, no. 1, pp. 73–79, 1998.

[248] A. R. Mushegian and E. V. Koonin, "A minimal gene set for cellular life derived by comparison of complete bacterial genomes," *Proceedings of the National Academy of Sciences*, vol. 93, no. 19, pp. 10 268–10 273, 1996.

[249] F. Arigoni, F. Talabot, M. Peitsch, M. D. Edgerton, E. Meldrum, E. Allet, R. Fish, T. Jamotte, M. L. Curchod, and H. Loferer, "A genome-based approach for the identification of essential bacterial genes." *Nat Biotechnol*, vol. 16, no. 9, pp. 851–856, Sep. 1998.

[250] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool." *Journal of molecular biology*, vol. 215, no. 3, pp. 403–410, Oct. 1990.

[251] M. Seringhaus, A. Paccanaro, A. Borneman, M. Snyder, and M. Gerstein, "Predicting essential genes in fungal genomes," *Genome Research*, vol. 16, no. 9, pp. 1126–1135, 2006.

[252] A. Gustafson, E. Snitkin, S. Parker, C. DeLisi, and S. Kasif, "Towards the identification of essential genes using targeted genome sequencing and comparative analysis," *BMC Genomics*, vol. 7, no. 1, p. 265, 2006.

[253] H. Jeong, S. P. Mason, A.-L. Barabasi, and Z. N. Oltvai, "Lethality and centrality in protein networks," *NATURE*, vol. 411, p. 41, 2001.

[254] H. Wang, M. Li, J. Wang, and Y. Pan, "A new method for identifying essential proteins based on edge clustering coefficient," in *Bioinformatics Research and Applications*, ser. Lecture Notes in Computer Science, J. Chen, J. Wang, and A. Zelikovsky, Eds. Springer Berlin / Heidelberg, 2011, vol. 6674, pp. 87–98.

[255] C.-C. Lin, H.-F. Juan, J.-T. Hsiang, Y.-C. Hwang, H. Mori, and H.-C. Huang, "Essential core of protein-protein interaction network in escherichia coli," *Journal of Proteome Research*, vol. 0, no. 0, 2009.

[256] H. Liang and W.-H. Li, "Gene essentiality, gene duplicability and protein connectivity in human and mouse," *Trends in Genetics*, vol. 23, no. 8, pp. 375 – 378, 2007.

[257] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, Mar. 1977.

[258] M. P. P. Joy, A. Brock, D. E. Ingber, and S. Huang, "High-betweenness proteins in the yeast protein interaction network." *J Biomed Biotechnol*, vol. 2005, no. 2, pp. 96–103, 2005.

[259] S. Wuchty and P. F. Stadler, "Centers of complex networks." *J Theor Biol*, vol. 223, no. 1, pp. 45–53, Jul. 2003.

[260] E. Estrada and J. A. Rodríguez-Velázquez, "Subgraph centrality in complex networks." *Phys Rev E Stat Nonlin Soft Matter Phys*, vol. 71, no. 5 Pt 2, May 2005.

[261] P. Bonacich, "Power and centrality: A family of measures," *The American Journal of Sociology*, vol. 92, no. 5, pp. 1170–1182, 1987.

[262] E. Estrada, "Virtual identification of essential proteins within the protein interaction network of yeast." *Proteomics*, vol. 6, no. 1, pp. 35–40, Jan. 2006.

[263] M. Li, J. Wang, X. Chen, H. Wang, and Y. Pan, "A local average connectivity-based method for identifying essential proteins from the network level," *Computational Biology and Chemistry*, Apr. 2011.

[264] S. $http://www-sequence.stanford.edu/group/yeast\_deletion\_project/deletions3.html$, "Saccharomyces genome deletion project."

[265] K. Park and D. Kim, "Localized network centrality and essentiality in the yeast-protein interaction network," *PROTEOMICS*, vol. 9, no. 22, pp. 5143–5154, 2009.

[266] G. del Rio, D. Koschutzki, and G. Coello, "How to identify essential genes from molecular networks?" *BMC Systems Biology*, vol. 3, no. 1, p. 102, 2009.

[267] A. Martinez-Antonio and J. Collado-Vides, "Identifying global regulators in transcriptional regulatory networks in bacteria," *Current Opinion in Microbiology*, vol. 6, no. 5, pp. 482 – 489, 2003.

[268] F. Provost, T. Fawcett, and R. Kohavi, "The Case Against Accuracy Estimation for Comparing Induction Algorithms," in *In Proceedings of the Fifteenth International Conference on Machine Learning*, 1997, pp. 445–453.

[269] C. D. Manning and H. Schütze, *Foundations of statistical natural language processing*. Cambridge, MA, USA: MIT Press, 1999.

[270] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 233–240.

[271] J. D. E. Burnside, R. Ramakrishnan, V. S. Costa, and J. Shavlik, "View learning for statistical relational learning: With an application to mammography," in *Proceeding of the 19th International Joint Conference on Artificial Intelligence*, 2005, pp. 677–683.

[272] P. Singla and P. Domingos, "Discriminative training of markov logic networks," in *In Proc. of the Natl. Conf. on Artificial Intelligence*, 2005.

[273] S. Cole, "Comparative mycobacterial genomics as a tool for drug target and antigen discovery," *European Respiratory Journal*, vol. 20, no. 36 suppl, pp. 78s–86s, 2002.

[274] G. T. Hart, A. Ramani, and E. Marcotte, "How complete are current yeast and human protein-interaction networks?" *Genome Biology*, vol. 7, no. 11, p. 120, 2006.

[275] G. Lamichhane, M. Zignol, N. J. Blades, D. E. Geiman, A. Dougherty, J. Grosset, K. W. Broman, and W. R. Bishai, "A postgenomic method for predicting essential genes at subsaturation levels of mutagenesis: Application to mycobacterium tuberculosis," *Proceedings of the National Academy of Sciences*, vol. 100, no. 12, pp. 7213–7218, 2003.

[276] H. Jeong, Z. N. Oltvai, and A.-L. Barabasi, "Prediction of protein essentiality based on genomic data," *Complexus*, vol. 1, no. 1, pp. 19–28, 2003.

[277] E. R. DeLong, D. M. DeLong, and D. L. Clarke-Pearson, "Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach." *Biometrics*, vol. 44, no. 3, pp. 837–845, Sep. 1988.

[278] B.-J. Breitkreutz, C. Stark, T. Reguly, L. Boucher, A. Breitkreutz, M. Livstone, R. Oughtred, D. H. Lackner, J. Bähler, V. Wood, K. Dolinski, and M. Tyers, "The BioGRID interaction database: 2008 update," *Nucleic Acids Research*, vol. 36, no. suppl 1, pp. D637–D640, Jan. 2008.

[279] S. Visa, "Issues in mining imbalanced data sets - a review paper," in *Proceedings of the Sixteen Midwest Artificial Intelligence and Cognitive Science Conference, 2005*, 2005, pp. 67–73.

[280] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *SIGKDD Explorations*, vol. 11, no. 1, pp. 10–18, Nov. 2009.

[281] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Trans Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, Mar. 1998.

[282] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations (The Morgan Kaufmann Series in Data Management Systems)*, 1st ed. Morgan Kaufmann, Oct. 1999.

[283] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[284] J. R. Quinlan, *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*, 1st ed. Morgan Kaufmann, Oct. 1992.

[285] H. Shi, "Best-first decision tree learning," University of Waikato, Tech. Rep., 2007.

[286] N. Landwehr, M. Hall, and E. Frank, "Logistic model trees," *Machine Learning*, pp. 161–205, May 2005.

[287] Y. Freund and L. Mason, "The alternating decision tree learning algorithm,," in *Proc. 16th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 1999, pp. 124–133.

[288] J. C. Platt, *Fast training of support vector machines using sequential minimal optimization*. Cambridge, MA, USA: MIT Press, 1999, pp. 185–208.

[289] D. W. Ruck, S. K. Rogers, M. Kabrisky, M. E. Oxley, and B. W. Suter, "The multilayer perceptron as an approximation to a bayes optimal discriminant function," *IEEE Transactions on Neural Networks*, vol. 1, no. 4, pp. 296–298, Dec. 1990.

[290] R. Kohavi, "Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, E. Simoudis, J. W. Han, and U. Fayyad, Eds. AAAI Press, 1996, pp. 202–207.

[291] I. A. Vergara, T. Norambuena, E. Ferrada, A. W. Slater, and F. Melo, "StAR: a simple tool for the statistical comparison of ROC curves," *BMC Bioinformatics*, vol. 9, p. 265, Jun. 2008.

[292] G. del Rio, D. Koschützki, and G. Coello, "How to identify essential genes from molecular networks?" *BMC systems biology*, vol. 3, no. 1, pp. 102+, 2009.

[293] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.

[294] D. A. Ross and R. S. Zemel, "Learning parts-based representations of data," *J. Mach. Learn. Res.*, vol. 7, pp. 2369–2397, 2006.

[295] D. D. Lee and H. S. Seung, "Unsupervised learning by convex and conic coding," in *Advances in Neural Information Processing Systems 9*. MIT Press, 1997, vol. 9, pp. 515–521.

[296] D. Lee and H. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[297] D. Donoho and V. Stodden, "When does non-negative matrix factorization give a correct decomposition into parts," in *Advances in Neural Information Processing Systems 16*, 2004.

[298] S. Z. Li, X. Hou, H. Zhang, and Q. Cheng, "Learning spatially localized, parts-based representation," in *CVPR '01: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. 207–212.

[299] V. P. Pauca, J. Piper, and R. J. Plemmons, "Nonnegative matrix factorization for spectral data analysis," *Linear Algebra and Its Applications*, vol. 416, no. 1, pp. 29–47, 2006.

[300] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. New York, NY, USA: ACM Press, 2003, pp. 267–273.

[301] J. Brunet, P. Tamayo, T. Golub, and J. Mesirov, "Metagenes and molecular pattern discovery using matrix factorization," *Proceedings of the National Academy of Sciences*, vol. 101, no. 12, pp. 4164–4169, 2004.

[302] P. O. Hoyer, "Non-negative sparse coding," in *Proc. IEEE Workshop on Neural Networks for Signal Processing*, 2002, pp. 557–565.

[303] ——, "Non-negative matrix factorization with sparseness constraints," *The Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[304] Y. Gao and G. Church, "Improving molecular cancer class discovery through sparse non-negative matrix factorization," *Bioinformatics*, vol. 21, no. 21, pp. 3970–3975, 2005.

[305] D.P.Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA 02178-9998: Athena Scientific, 1999.

[306] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems 13*. MIT Press, 2001, pp. 556–562.

[307] C. L. LAWSON and R. J. HANSEN, *Solving Least Squares Problems*. Englewood Cliffs, N.J., USA: Prentice-Hall, 1974.

[308] A. Björck, *Numerical Methods for Least Squares Problems*. SIAM: Society for Industrial and Applied Mathematics, 1996.

[309] R. Bro and S. D. Jong, "A fast non-negativity-constrained least squares algorithm," in *Journal of Chemometrics*, vol. 11, 1997, pp. 393–401.

[310] R. Harshman and M. Lundy, "The PARAFAC model for three-way factor analysis and multidimensional scaling, in research methods for multimode data analysis," 1984.

[311] M. V. Benthem and M.R.Keenan, "Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems," in *Journal of Chemometrics*, vol. 18, 2004, pp. 441–450.

[312] R. C. Taylor, "An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics." *BMC bioinformatics*, vol. 11 Suppl 12, no. Suppl 12, pp. S1+, 2010.

Appendix A

NONNEGATIVE AND BOUNDED MATRIX FACTORIZATION

This chapter describes a nonnegative matrix factorization (NMF) and a bounded matrix factorization (BMF) which is a generalized NMF algorithm. Matrix factorization algorithm was originally introduced for dimension reduction, and many applications have used the method for clustering task as well. We first review NMF algorithm and a sparse nonnegative matrix factorization (SNMF) which is a specification of NMF algorithm used for data clustering. The SNMF algorithms are used to find large-scale protein motifs in Chapter 3 and to cluster a network in Chapter 4 as NMF-BNM and NMFGO-BNM algorithms. As the name indicates, NMF is applicable only for nonnegative data. For a generalized NMF algorithm which can be applied to negative values as well, we introduce bounded matrix factorization (BMF) algorithm in the following section.

A.1 Nonnegative Matrix Factorization

Nonnegative matrix factorization (NMF), first introduced as a positive matrix factorization (PMF) by Paatero and Tapper [293], is a matrix analysis that has attracted much attention during the past decade. Besides NMF, there are several matrix factorization methods used in many applications, including principal component analysis (PCA) and vector quantization (VQ). All of those matrix factorization methods represent the data as vectors and form a data matrix to decompose it into two factor sub-matrices. Ross and Zemel [294] noted that when data are represented as vectors, parts of the data are interpreted with subsets of the bases that take on values in a coordinated fashion. Although other factorization methods are related to this interpretation in general, only NMF has a sparse and part-based localization property [295, 296], but under special conditions [297]. NMF is considered for high dimensional data where each entry has a nonnegative value, and it provides a lower rank approximation formed by factors whose entities are also nonnegative. NMF was successfully applied to analyzing face images [296, 298], text corpus [299], and many other

tasks in computational biology [214]. Areas of application include molecular pattern discovery, class prediction, functional analysis of genes, and biomedical informatics.

Given an $m \times n$ data matrix A, nonnegative factors such as W, H are commonly computed by solving the following objective function,

$$\min_{W,H} \frac{1}{2}\|A - WH\|_F^2 \text{ s.t. } W \geq 0, H \geq 0, \tag{A.1}$$

where W is the $m \times k$ bases matrix, H is the $k \times n$ coefficient matrix, and k is usually much smaller than $\min(m, n)$. The interpretation of factored matrices, W and H, depend on the domain of application. For instance, if the data matrix A denotes microarray data, the rows correspond to expression levels of genes and the columns correspond to samples representing distinct tissues, experiments, or time points. Thus, $A(i, j)$ describes the i^{th} gene expression level for the j^{th} sample. If the microarray data matrix A is factored into W and H using NMF, each column of W defines a metagene and each column of H represents the metagene expression pattern of the corresponding sample. In this case, the metagenes of W summarize gene behaviors across samples, while the patterns of H summarize the behavior of samples across genes. On the other hand, if it is for data clustering, each basis of W can represent a prototype of each cluster and each column of H is the relevance of the data sample corresponding to each prototype.

The nonnegativity of W and H provides a pleasing interpretation of the factorization. Each object is explained by an additive linear combination of intrinsic 'parts' of the data [296]. This property of NMF gives an intuitive meaning and physical interpretation, especially for large-scale data, while the orthogonal components with arbitrary signs in PCA lack their conceptual interpretation. In face image applications with NMF [296], the column vectors of W represent each component of the face, that is, nose, eyes, cheeks, etc. In addition to the natural interpretability as a dimension reduction method, NMF has shown favorable performance for clustering tasks. For text clustering, Xu *et al.* [300] reported competitive performance of NMF compared to other methods in spectral clustering. Brunet *et al.* [301] used NMF on cancer microarray data and demonstrated its ability to detect cancer classes.

A.2 Sparse Nonnegative Matrix Factorization

Generally, NMF provides sparse and part-based representations, but this may not always be the case. Li *et al.* [298] and Hoyer [302] presented part-based but holistic (non-local) representations produced by NMF. These results exemplify that nonnegativeness is an insufficient condition to produce sparse representations. Therefore, many studies [298, 302–304] focused on enforcing sparseness explicitly on W, H or both. H. Kim and Park [186, 213] proposed a sparse NMF (SNMF) using a refined formulation with an additional penalty term and proposed an efficient algorithm. SNMF was further studied by J. Kim and Park [215], where they demonstrated that SNMF gives more consistent clustering results than a K-means algorithm.

In this thesis, we use SNMF with sparseness enforced on the right factor H (SNMF/R) used by H. Kim and Park in [186] to cluster a set of protein segments in Chapter 3 and a PPI network in Chapter 4. We note that H. Kim et al. [186, 213] provided the SNMF with sparseness enforced on the left factor W (SNMF/L) as well, which is useful for representing part-based bases, but not for a clustering application. We provide an objective function for SNMF/R as the following.

Given a nonnegative matrix A, find nonnegative matrix factors W and H such that;

$$\min_{W,H} \frac{1}{2}\{\|A - WH\|_F^2 + \eta\|W\|_F^2 + \beta \sum_{j=1}^{m} \|H(:, j)\|_1^2\} \tag{A.2}$$

$$\text{subject to } W \geq 0, H \geq 0.$$

where $\|.\|_F^2$ is the square of the Frobenius norm, $\|.\|_1^2$ of the L_1 norm, and $H(:, j)$ is the j^{th} column of matrix H. Regularization using L_1-norm promotes sparse solutions on the factor H. Two parameters, η and β, are involved, where η suppresses the Frobenius norm of W, and β regulates balances between the sparseness of matrix H and the accuracy of the factors. In practice, the parameters are adjusted empirically as they are affected by the data size and the number of clusters. Alternating nonnegativity constrained least squares algorithm using the active set method [186] was used to obtain W and H. By arranging n number of proteins to column-wise, we form an $m \times n$ data matrix A. After deciding the number of clusters k, we use alternating nonnegativity

constrained least square algorithm, and factor A into W and H factor matrices. Each column of H is k-dimensional vector, where i^{th} entry represents the relevance of i^{th} cluster of the corresponding sample. Each data is then assigned to the cluster of maximum relevance.

A.3 Bounded Matrix Factorization

We develop a bounded matrix factorization (BMF) algorithm as an generalization of non-negative matrix factorization(NMF). NMF has been shown to be useful in many applications with multivariate data. However, some applications require the decomposed data lie within specific boundaries, which can include negative values as well.

Bounded matrix factorization (BMF), therefore, is motivated to utilize those bounded, or possibly negative data and factorize it into meaningful factors. BMF is an extension of NMF as NMF is easily specified with BMF if the lower boundary is set to zero and upper boundary to infinity. In this research, we assume that the lower and upper bounds are uniform as we can extend it to varying boundaries later. We mostly follow the framework of [213] by H. Kim and Park where the two block coordinate decent method is proposed and an alternating nonnegativity constrained least squares and the active set method [305] is used for NMF. For BMF, we use an alternating *bounded* least squares instead of *nonnegativity* constrained least squares.

BMF is considered for high dimensional data where each component is bounded, and it provides a lower rank approximation formed by factors whose elements are also bounded. Given an $m \times n$ matrix A , the factors of W and H are computed by minimizing

$$\frac{1}{2} \parallel A - WH \parallel_F^2, s.t.\ \ l_h \leq H \leq u_h, l_w \leq W \leq u_w \tag{A.3}$$

Here, W is an $m \times k$ bases matrix and H is $k \times n$ coefficient matrix where $k << min(m, n)$. It is clear that BMF is a specific type of NMF as the Equation (A.3) is same as Equation (A.1) if we set $l_h = l_w = 0$ and $u_h = u_w = \infty$.

A.3.1 Two Block Coordinate Descent Framework for BMF

We can apply a two block coordinate descent problem [305] to the algorithm of BMF, as it has been a basic frame work of NMF as well. Given an $m \times n$ matrix A, we alternate the following alternating bounded constrained least squares of Equation (A.4) and (A.5)until a convergence criterion is satisfied;

$$min_W \parallel H^T W^T - A^T \parallel_F^2, s.t.\ l_w \leq W \leq u_w, \tag{A.4}$$

where H is fixed and

$$min_H \parallel WH - A \parallel_F^2, s.t.\ l_h \leq H \leq u_h, \tag{A.5}$$

where W is fixed.

Since the two subproblems are symmetric, we can start from initializing anyone of the two factors. If H is initialized first, then W is obtained with Equation (A.4). H is then updated with Equation (A.5), and the two updates are repeated until a stopping criterion is satisfied.

One of a stopping criterion is when we reach stationary points of H and W as the BMF problem of Equation (A.3) is non-convex like the NMF problem. H. Kim and Park [213] used the Karush-Kuhn-Tucker Conditions [305] to find a stationary point for the NMF problem, and we also applied the KKT conditions for BMF. Even with non-convex formulation of the two subproblems of Equation (A.4) and (A.5), any limit points can be a stationary point. Therefore, (W, H) is a stationary point of Equation (A.3) if and only if the following conditions meet.

$$\begin{array}{llll}
l_w \leq W \leq u_w & & , & l_h \leq H \leq u_h \\
\bigtriangledown_W f(W) = 0 & \text{where} \quad l_w < W < u_w & , & \bigtriangledown_H f(H) = 0 \quad \text{where} \quad l_h < H < u_h \\
\bigtriangledown_W f(W) > 0 & \text{where} \quad W = l_w & , & \bigtriangledown_H f(H) > 0 \quad \text{where} \quad H = l_h \\
\bigtriangledown_W f(W) < 0 & \text{where} \quad W = u_w & , & \bigtriangledown_H f(H) < 0 \quad \text{where} \quad H = u_h
\end{array}$$

These conditions are rewritten as

$$min(W - l_w, \bigtriangledown_W f(W)) = 0 \quad \text{and} \quad min(u_w - W, \bigtriangledown_W f(W)) = 0$$
$$min(H - l_h, \bigtriangledown_H f(H)) = 0 \quad \text{and} \quad min(u_h - H, \bigtriangledown_H f(H)) = 0$$

In the process when a factor is updated while the other factor is fixed, Lee and Seung [306] suggested the norm-based multiplicative update rules and the divergence-based multiplicative update rules for NMF. However, the former rule is inapplicable when the factors are zero, and the latter rule needs well-defined objective function. Therefore, we update each factor based on alternating bounded least squares (BLS) and the active set method, motivated by the work of H.Kim and Park [213].

A.3.2 BLS based on the Active Set method

As the NMF/ANLS [213] is based on the nonnegative least squares problems (NNLS), we can derive a bounded matrix factorization based on the alternating bounded least squares and the active set method. Hence, we review the algorithm of bounded least square (BLS) problems in this section, then extend it to the bounded matrix factorization (BMF).

Both of the NNLS and BLS are treated as the special cases of least squares with inequality constraints (LSI) in [307]. Given a two matrix W and A, the problem of LSI is finding a vector h satisfying the following;

$$min_h \|Wh - A\| \text{ subject to } \; l \leq Ch \leq u. \tag{A.6}$$

Lawson and Hanson [307] used ***active set algorithms*** which are iterative in nature. An ***active set*** consists of the indices of vector h where constraints are satisfied with equality, that is, on the boundary. If the *true* active set is known, then the solution would be the same as that with equality constraints only, that is, the problem of least square equality (LSE). Hence, active set algorithm is a sequence of LSEs with respect to predicted active set at current iteration. The difference of Wh and A in Equation (A.6) decreases at each iteration, and it will eventually find an optimal point.

In fact, the active set algorithm has two phases; a phase for the feasibility and a phase for the optimality of the current estimates. A point h is called feasible when it satisfies the constraints. Lawson and Hansen [307] consider NNLS as a special cases of LSI and they provided a detail ***active set algorithm*** for the problem NNLS, which is defined as;

$$min_h \|Wh - A\| \text{ subject to } \quad h \geq 0. \tag{A.7}$$

As a further step, Björck presented an algorithm for the least squares with bounded constraints, that is, the problem BLS in detail [308], and called it an ***active set algorithm for problem BLS.*** Again, BLS is a special case of the problem of LSI with simple boundaries of l and u, as the following;

$$min_h \|Wh - A\| \text{ subject to } \quad l \leq h \leq u. \tag{A.8}$$

The basic idea for NNLS and BLS is the same. Here, we focus only on explaining the algorithm for BLS presented in [308]. Although general active set algorithms use different terms, here we will use ***fixed*** (or ***bounded***) and ***free*** set instead of ***active*** and ***passive*** set respectively, to make the algorithm clearer.

The active set algorithm for BLS is stated as the followings; Given a matrix $W \in \mathbf{R^{m \times n}}$ and $a \in \mathbf{R^m}$, the active set algorithm for problem BLS is finding a vector $h \in \mathbf{R^n}$ minimizing an objective function $f(h) = \|Wh - a\|^2$ subject to $l \leq h \leq u$. The index set of h is divided into the ***free*** set ($\mathcal{F}$) and ***fixed*** or ***bounded*** set ($\mathcal{B}$).

$$\{1, 2, \cdots, n\} = \mathcal{F} \bigcup \mathcal{B},$$

Here $i \in \mathcal{F}$ if h_i is a free variable within the (strict) boundary and $i \in \mathcal{B}$ if it is a fixed at its lower (l) or upper bound (u).

An initial feasible vector is found as the median of the lower and upper bound values. Free set and fixed set are also initialized with respect to the initial solution h, that is, $\mathcal{F} = \{1, 2, \cdots, n\}$ and $\mathcal{B} = \emptyset$. Then the algorithm repeats two alternating processes for optimality and feasibility of the solution until it converges. After an optimal solution z without constraints is obtained, the

Algorithm 6: BLS(W, l, u, a)

input : Matrix $W \in R^{m \times n}$ and vector $a \in R^m$.
output a vector $h \in R^n$, subject to $l \geq h \geq u$
:

Let $h := (l + u)/2$
$\mathcal{F} := \{1, 2, \cdots, n\}$
$\mathcal{B} := \phi$
Compute $z = argmin_h \parallel Wh - a \parallel$.
while $\exists\, i \in \mathcal{F}$ *such that* $z_i < l$ *or* $z_i > u$ **do**
    $\forall i \in \mathcal{F}$ such that z_i is infeasible: $\alpha_i = \begin{cases} (h_i - l)/(h_i - z_i), & z_i < l \\ (u - h_i)/(z_i - h_i); & z_i > u \end{cases}$
    $\alpha := min_{i \in \mathcal{F}} \alpha_i; \quad h := h + \alpha(z - h); \quad (0 \leq \alpha < 1)$.
    Move from $\mathcal{F}$ to $\mathcal{B}$ all indices j for which $h_j = l$ or $h_j = u$.
    Let $\gamma = \begin{cases} \gamma_j := 1, & \text{if } h_j = l \\ \gamma_j := -1, & \text{if } h_j = u \end{cases}$
    Compute Lagrange multipliers λ with respect to the boundary components.
    while $\exists i \in \mathcal{B}$ *such that* $sign(\gamma_i)\lambda_i > 0$ **do**
        Find index t such that $sign(\gamma_t)$.
        Move index t from $\mathcal{B}$ to $\mathcal{F}$.
        Compute $z_{\mathcal{F}} = argmin_h \parallel W_{\mathcal{F}} h - a \parallel$.
        $h_{\mathcal{F}} = z_{\mathcal{F}}$.
        Compute Lagrange multipliers λ with respect to the current boundary components.
    Compute $z_{\mathcal{F}} = argmin_h \parallel W_{\mathcal{F}} h - a \parallel$, the optimal solution with respect to free components.
$h_{\mathcal{F}} = z_{\mathcal{F}}$.

free indices of z are checked for the feasibility. If the current solution z is infeasible with respect to the current free indices, a new h is derived on the segment line between z and an old h. The optimality of h is checked using Lagrange multiplier ($\lambda = W^T(a - Wh)$) on the boundaries. This process approaches toward minimizing the objective function and the convergence of the algorithm is supported by the KKT condition.

In the Algorithm 6, most of computational cost comes from computing z which minimizing $\|Wh - a\|$ and several implementations are available. Noting that indices of h relate to the columns of W, the basic idea is re-arranging the columns of W so that the indices of free set are grouped into forward.

Let;

$$W_{\mathcal{F}} := \begin{cases} \text{column } j \text{ of } W & \text{if } j \in \mathcal{F} \\ 0 & \text{if } j \in \mathcal{B} \end{cases} \tag{A.9}$$

$$W_{\mathcal{B}} := \begin{cases} \text{column } j \text{ of } W & \text{if } j \in \mathcal{B} \\ 0 & \text{if } j \in \mathcal{F} \end{cases} \tag{A.10}$$

Then $z_{\mathcal{F}} = W_{\mathcal{F}}^{+}(a - W_{\mathcal{B}}h_B)$ is an optimal solution with respect to $\mathcal{F}$, where $W_{\mathcal{F}}^{+}$ is a pseudo-inverse of $W_{\mathcal{F}}$.

In most cases for BLS, the matrix W is over-determined. Hence, it is efficient to use a QR decomposition. Björck specifically used QR decomposition with column permutation for efficient computing [308]. He defines a matrix $E_{\mathcal{F}} \in \mathbf{R}^{\mathbf{n}\times\|\mathcal{F}\|}$ consisting of the columns e_i^T, $i \in \mathcal{F}$ and $E_{\mathcal{B}}$ with respect to $\mathcal{B}$ as well. Then $W(E_{\mathcal{F}}, E_{\mathcal{B}})$ divides W into $W_{\mathcal{F}}$ and $W_{\mathcal{B}}$. If we apply QR decomposition to the matrix $W(E_{\mathcal{F}}, E_{\mathcal{B}})$, $Q^T W(E_{\mathcal{F}}, E_{\mathcal{B}}) = \begin{pmatrix} R & S \\ 0 & U \end{pmatrix}$, $Q^T a = \begin{pmatrix} c \\ d \end{pmatrix}$ where the rank of R is same as $\|\mathcal{F}\|$. This way we can compute $z_{\mathcal{F}} = R^{-1}(c - SE_{\mathcal{B}}^T h)$ more efficiently then computing the pseudo-inverse of $W_{\mathcal{F}}$.

The QR decomposition gives other advantages to the algorithm. Besides of the fact that QR is more efficient than computing the pseudo-inverse of W, it can be efficiently updated as only one column is removed or added at each step. The efficient QR updating after one column vector is added or removed is described in [307] at chapter 24. The QR decomposition also reduces the computing of Lagrange multiplier $\lambda = W^T(a - Wh)$, as it is obtained with $\lambda = U^T(d - UE_{\mathcal{B}}^T h)$, where U is defined in the BLS, shown in Algorithm 6.

A.3.3 Fast Combinatorial Bounded Least Squares

We move on the problem of BLS with multiple right hand side (RHS) vectors, that is, finding H minimizing $\|WH - A\|_2$. If this is the case with unconstrained least squares, then $H_{LS} = W^+ A$ and computing the pseudo-inverse of W would be efficient enough. With constraints, however, this means we have to compute pseudo-inverse of $W_{\mathcal{F}}$s for each column of A. Although, it seems legitimate to solve problem BLS for each column sequentially, treating each problem independently,

this approach is very inefficient when the number of RHS vectors is extremely large. In addition, it can be very redundant, when same elements of the fixed sets appear repeatedly from column to column.

Therefore, for efficient computing with large number of columns in A, the challenge in the problem of BLS with multiple RHS is finding any common factor throughout the columns to reduce any repeating and redundant calculations.

As shown in Algorithm 6 for single RHS vector, the main computations come from the two parts; computing $z_{\mathcal{F}}$,

$$z_{\mathcal{F}} = (W_{\mathcal{F}}^T W_{\mathcal{F}})^{-1} W_{\mathcal{F}}^T a \tag{A.11}$$

and computing Lagrange multiplier at each iteration.

$$\lambda = W^T(a - Wh) \tag{A.12}$$

With n number or RHS vectors, Equation (A.11) and (A.12) need to be repeated for n times, in sequential computing methodology. Equation (A.11) is in fact much expensive as the $\mathcal{F}$ changes even with one RHS vector. However, it is quite possible that many of the single RHS problems share the same elements in their free (active) sets $\mathcal{F}$. In that case, computing the pseudo-inverse of $W_{\mathcal{F}}$ separately is redundant. There should be a way to save the common elements in free sets and re-use the pre-computed factors.

For the efficient computing of problem NNLS with multiple RHS vectors, Bro and DeJong presented a fast NNLS (FNNLS) algorithm [309] by applying PARAFAC model [310]. Based on the FNNLS algorithm, Benthem and Keenan provided a more efficient algorithm for problem NNLS in [311], called it fast combinatorial NNLS (FC-NNLS) algorithm, as an *column-parallel* computing.

FNNLS algorithm modified the Equation (A.11) as

$$z_{\mathcal{F}} = ((W^T W)_{\mathcal{F}})^{-1} (W^T a)_{\mathcal{F}} \tag{A.13}$$

and (A.12) as

$$\lambda = W^T a - W^T W h, \tag{A.14}$$

where $(W^T W)_{\mathcal{F}}$ is the notation indicating that it is the sub-matrix of $W^T W$ containing only the rows and columns corresponding to $\mathcal{F}$. In this way, we can reveal the constant parts, which is, $W^T W$. Bro and Dejong emphasize that this also improves the speed, since most case of problem NNLS, W has more rows then columns, therefore $W^T W$ is much smaller than W. They also initialize H as the solution of unconstraint least squares to force only small changes allowed during the iterations, for more speed up. FC-NNLS [311] further improved the speed with the algorithm of combinatorial subspace least squares (CSSLS), which is the engine of FC-NNLS, identifying and grouping *unique* free sets among the columns.

We modify the FC-NNLS algorithm to adjust it for problem BLS , and call it FC-BLS algorithm. The FC-BLS also uses algorithm CSSLS as an engine for speed up. The FC-BLS follows most of the process of FC-NNLS [311], except the optimal solution $z_{\mathcal{F}}$ in the free set $\mathcal{F}$. With the observations of the followings, we can derive a similar equation to Equation (A.13):

If the free set $\mathcal{F}$ and fixed set $\mathcal{B}$ are given, we can define $W_{\mathcal{F}}$ and $W_{\mathcal{B}}$ described in Equation (A.9) and (A.10), then we re-arrange the columns of W such as, $W = \begin{pmatrix} W_{\mathcal{F}} & W_{\mathcal{B}} \end{pmatrix}$. Likewise, the current solution h is divided into $\begin{pmatrix} h_{\mathcal{F}} \\ h_{\mathcal{B}} \end{pmatrix}$. Then z with respect to the free set $\mathcal{F}$ is; $z_{\mathcal{F}} = W_{\mathcal{F}}^{+}(a - W_{\mathcal{B}} h_{\mathcal{B}})$, where $W_{\mathcal{F}}^{+} = (W_{\mathcal{F}}^T W_{\mathcal{F}})^{-1} W_{\mathcal{F}}^T$.

If we reveal the constant parts,

$$\begin{aligned} z_{\mathcal{F}} &= [(W^T W)_{\mathcal{F}}]^{-1}(W^T a)_{\mathcal{F}} - [(W^T W)_{\mathcal{F}}]^{-1}\ {}_{\mathcal{F}}(W^T W)_{\mathcal{B}}\, h_{\mathcal{B}} \\ &= [(W^T W)_{\mathcal{F}}]^{-1}((W^T a)_{\mathcal{F}} -\ {}_{\mathcal{F}}(W^T W)_{\mathcal{B}}\, h_{\mathcal{B}}), \end{aligned}$$

where ${}_{\mathcal{F}}(W^T W)_{\mathcal{B}}$ is the sub-matrix of $W^T W$ consisting the rows of $\mathcal{F}$ and columns of $\mathcal{B}$.

Therefore, after we compute $W^T W$ and $W^T a$ only once, the corresponding sub-matrices are formed by choosing the corresponding rows and columns only whenever the free and fixed sets are updated. Algorithm 7 summarizes the process of FC-BLS, and CSSLS algorithm is reviewed in

Algorithm 8 as well since it is used as a subroutine of FC-BLS. For details of algorithm CSSLS, see [311].

Algorithm 7: FC-BLS(W, A, Z)

input : Matrix $A \in R^{m\times n}$, $W \in R^{m\times k}$ and $Z \in R^{k\times n}$.
output A matrix $H \in R^{k\times n}$.
:
$\mathcal{M} = \{1, 2, \cdots, k\}$
$\mathcal{N} = \{1, 2, \cdots, n\}$
Pre-compute W^TW and W^TA
$H = agrmin \parallel WH - A \parallel_2$
$H_{ij} = l$, if $H_{ij} < l$. $H_{ij} = u$, if $H_{ij} > u$.
$\Gamma_{ij} = 1$ if $H_{ij} = l$
$\Gamma_{ij} = -1$ if $H_{ij} = u$
$F = \{f_1\ f_2 \cdots\ f_k\}$ where $f_j = \{i \in \mathcal{N} : l < H_{ij} < u\}$
$B = \sim F$
$\mathcal{C} = \{j \in \mathcal{M} : F_j \neq \mathcal{N}\}$
Let $Z = H$ be the current solution.
while $\mathcal{C} \neq \emptyset$ **do**
    Compute $H_{\mathcal{C}} = argmin_{H_{\mathcal{C}}} \parallel WH_{\mathcal{C}} - A_{\mathcal{C}} \parallel$ using algorithm CSSLS and $F_{\mathcal{C}}$, Z
    $\mathcal{K} = \{j \in \mathcal{C} :\ if\ any\ (H_{ij}) < l\ or\ (H_{ij}) > u\}$
    $Z = H$.
    while $\mathcal{K} \neq \emptyset$ **do**
        $\forall h \in \mathcal{K}$, select the variables to move out of the free sets $\mathcal{F}_{\mathcal{K}}$
        Update solution $H_{\mathcal{K}}$ using algorithm CSSLS and $F_{\mathcal{K}}$ and current Z.
        Remove h from $\mathcal{K}$
    Check the optimality of solution $H_{\mathcal{C}}$
    Let $\mathcal{W}_{\mathcal{C}} := W^TA_{\mathcal{C}} - W^TWH_{\mathcal{C}}$.
    $\Lambda = \mathcal{W}. \times \Gamma$ %Multiplication of each element in the matrices.
    $\mathcal{J} = \Lambda_{\mathcal{C}} \leq 0$.
    $\mathcal{C} = \mathcal{C} - \mathcal{J}$ % Remove from $\mathcal{C}$ indices of columns whose solutions are optimal
    If $\mathcal{C} \neq \emptyset$, find $t = argmax_i(\Gamma_{\mathcal{C}}(i). \times \mathcal{W}_{\mathcal{C}}(i))$, where $i \in \mathcal{B}$. Move t from $\mathcal{B}$ to $\mathcal{F}$

The FC-BLS algorithm has initialization phase and main phase including two loops. The constant part of W^TW is pre-computed in the initialization phase. Also, the initial feasible solution H is computed as the solution of unconstraint LS problem using any efficient method, such as QR or LU decomposition. From the initial solution H, we identify the values between lower bound l and upper bound u, and save the indices to the free set $\mathcal{F}$. The values beyond the boundaries in H are overwritten by their close boundaries and the indices are saved to the fixed set $\mathcal{B}$. $\mathcal{C}$ consists

of the column indices whose solution are yet to be optimized. And the algorithm is repeated until $\mathcal{C} = \emptyset$.

The main phase is similar to that of BLS [308], except that FC-BLS is a column-parallel, and only the columns in $\mathcal{C}$ is considered. Given the current free and fixed set, compute the unconstrained solution Z and check its feasibility. Update H from the line segment of old H and Z if Z is infeasible. Otherwise, Z is checked for its optimality with respect to its free set. Details for the optimality and feasibility of BLS have been described in Algorithm 6.

As we can see in the summary of FC-BLS algorithm FC-BLS, it includes CSSLS algorithm as a subroutine. As aforementioned, this saves the computational cost further as it groups *unique* free sets in $\mathcal{F}$. The details is in the paper [311].

Algorithm 8: CSSLS$(W, A, \mathcal{F}, \mathcal{B})$

---

input : Matrix $A \in R^{m \times n}$, $W \in R^{m \times k}$, $\mathcal{F} \in R^{k \times n}$ free set and $\mathcal{B} \in R^{k \times n}$ be a fixed set
output A matrix $H \in R^{k \times n}$.
:

1 $H := Z$ %Initialize.
2 Let $\mathcal{M} = \{1, 2, \cdots, k\}$ %Index the columns of H.
3 $\mathcal{U} = \{\mathcal{U}_1, \mathcal{U}_2, \cdots, \mathcal{U}_t\}$ % *unique* free set.
4 $\mathcal{E}_j = \{i \in \mathcal{M} : \mathcal{F}_i = \mathcal{U}_j\}$. % Index of columns of H with identical Free sets.
5 $\forall j \in \{1, 2, \cdots, k\}$, $H_j = argmin_{l \leq H_{\mathcal{U}_j \mathcal{E}_j} \leq u} \parallel W_{\mathcal{U}_j \, \mathcal{U}_j} H_{\mathcal{E}_j} - A_{\mathcal{E}_j} \parallel$.
6 Return H.

Appendix B

PARALLEL NETWORK MOTIF SEARCH

B.1 Parallel search of network motifs

Network motifs in biological networks have been used in many applications in bioinformatics, including the typical patterns in different types of biological networks, prediction protein-protein interactions and generalizing network structure. The network motif detection process, however, requires high computation, as the number of searches grows exponentially with the size of network or the size of network motifs. Although approximation algorithms speed up the search time, results can be inconsistent. To overcome these limitations, we provide a parallel search algorithm where the works are easily distributed and scheduled.

First, we obtain a number of non-overlapped sub-networks through a network clustering algorithm, especially partition algorithm. Any algorithm introduced in chapter 4 is applicable to obtain these non-overlapped subnetworks. Through network clustering process, a number of boundary edges which connect different clusters are listed and defined as ***removed edges***. The subgraphs, called *missing subgraphs*, which will not searched in any clusters will be recovered from the removed edge list using RSRE (Recover Subgraphs from Removed Edges) algorithm. The search of missing subgraphs starting from individual edge will not be repeated with RSRE algorithm, which is also easily distributed to workers.

B.1.1 Recover Subgraphs from Removed Edges

We should note that after network clustering, there is a list of edges which do not belong to any of the clusters, called a list of removed edges (LRE). Figure B.1 shows an example original network and Figure B.2 shows the network after a clustering algorithm indicating the list of removed edges is $\{2, 10, 11, 14\}$. A key challenge is to recover the missing k-node subgraphs from this list without redundant counting. We provide an RSRE algorithm to enumerate all of the missing subgraphs

from the list of removed edges. Since this algorithm can be processed from each removed edge independently, we can apply query parallelization strategy. Query parallelization strategy is to search a whole network from each query point. Like ESU [15], the process of recovering missing subgraphs from LRE can be illustrated into a tree structure as shown in an example in Figure B.3, and each process can be parallelized.

Algorithm 9 illustrates the RSRE algorithm. For a given graph G, an integer k, and the list of removed edges LRE, RSRE enumerates all the missing subgraphs. We let PLE be the previously removed edges containing current edge in sequential process, or the edges whose label is less than or equal to that of the current edge. $EndPoints(ed)$ is the end vertices of ed and $EndPoints(S)$ for a set of edges S contains all the endpoints of all the edges of S. $N(e)$ is the set of neighbor edges of e and $N_{in}(S)$ is the set of neighbor edges including S itself. The algorithm starts with an edge e from LRE and E_{ext} set which is the neighbor edges with e, not containing PLE. Then in EXTENDFROME in Algorithm 10, an edge w from E_{ext} is added to form k-subgraph with $k-1$ edges with the following properties: The newly added w should introduce new vertices to V_S (see line 7). Also, adding w to E_{sub} should not involve an edge from PLE as shown in the line10. For example, in the Figure B.3, the edge-14 with root-10 cannot be added as the endpoints related to it are already in V_S. Likewise, the edge-9 cannot be added to the root edge-14 as it involves the edge-10 which is in PLE.

Algorithm 9: RSRE(G, k, LRE)

input : G, k, LRE
1 PLE $= \emptyset$
2 **while** *LRE* $\neq \emptyset$ **do**
3 Remove e from LRE
4 $PLE \leftarrow PLE \cup \{e\}$
5 $E_{ext} \leftarrow \{u \in N(e) | u \notin PLE\}$
6 $V_S \leftarrow EndPoints(e)$
7 EXTENDFROME$(\{e\}, V_S, E_{ext}, e, PLE, k)$.

Algorithm 10: EXTENDFROME$(E_{sub}, E_{ext}, rootEdge, k, PLE)$

input : $(E_{sub}, E_{ext}, rootEdge, k, PLE)$
output A number of k-sized subgraphs
:

$vertexset \leftarrow EndPoints(E_{sub})$
if $\mid vertexset \mid == k$ **then**
    output $vertexset$
while $E_{extend} \neq \emptyset$ **do**
    Remove w from E_{ext}
    if V_S *contains all* $EndPoints(w)$ **then**
        continue
    $subvertices \leftarrow vertices \cup EndPoints(w)$
    if $\exists ed \in G(subvertices)$ *where* $ed \in PLE$ **then**
        continue
    $V'_S \leftarrow V_S \cup EndPoints(w)$
    $N_{excl} \leftarrow \{a \in N(w) | a \notin \{PLE \cup N_{in}(E_{sub})\}\}$
    $E'_{sub} \leftarrow E_{sub} \cup \{w\}$
    $E'_{ext} \leftarrow N_{excl} \cup E_{ext}$
    EXTENDFROME$(E'_{sub}, V'_S, E'_{ext}, rootEdge, PLE, k)$.

B.2 Network clustering and parallel search

With a number of sub-networks and a list of removed edges, we can now exactly count all k-node subgraph to find network motifs in parallel. Parallel network motif search has been introduced in [21], but they allowed overlapping between sub-networks and repeated subgraphs are removed after they are searched, which requires redundant computing time.

We can use message passing interface (MPI) for this work. The parallel process has one master program and a number of workers. The master program cluster the original network and maintain job schedule for each worker. After obtaining a number of sub-networks and a list of removed edges, the master program assigns each sub-network and each edge from the list to each worker to exhaustively search k-node subgraphs, example is in Figure B.5. Based on the size of sub-network and the structure the network surrounding each edge, the computing time varies. The master program watches the job status of each worker and assigns another job to the worker if he

finished his previous job. It is easily maintainable and distributable as each sub-network and an edge from the list is independent.

Clustering algorithms not only help parallelization but also reduce the number of works if some of clusters are isomorphic graphs. For example, assume that we search 3-node subgraphs in a network of $G = (V, E)$ with $|V| = 16, |E| = 19$ of Figure B.1. After we apply a clustering algorithm, we obtain five clusters as shown in Figure B.2. However, the number of unique subgraphs is only three as the subgraph of Figure B.4 appears three times. Therefore, we search 3-node subgraphs from only one of them, and multiply it with three.

We believe the work in this thesis can help feasible computing for network motif finding as it can be easily parallelized without data-dependency. We are also interested in implementing the parallel network motifs search in cloud computing environment utilizing Hadoop MapReduce algorithm [312] in near future.

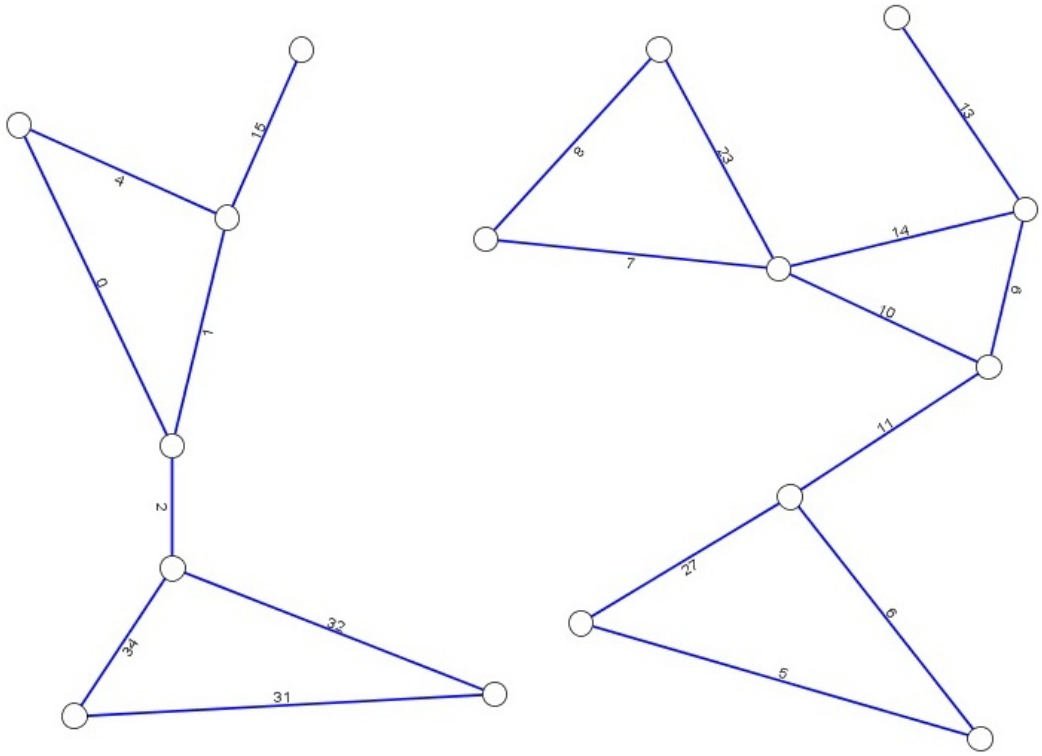


Figure B.1: An example network $G = (V, E)$ with $|V| = 16, |E| = 19$. This is an original network.

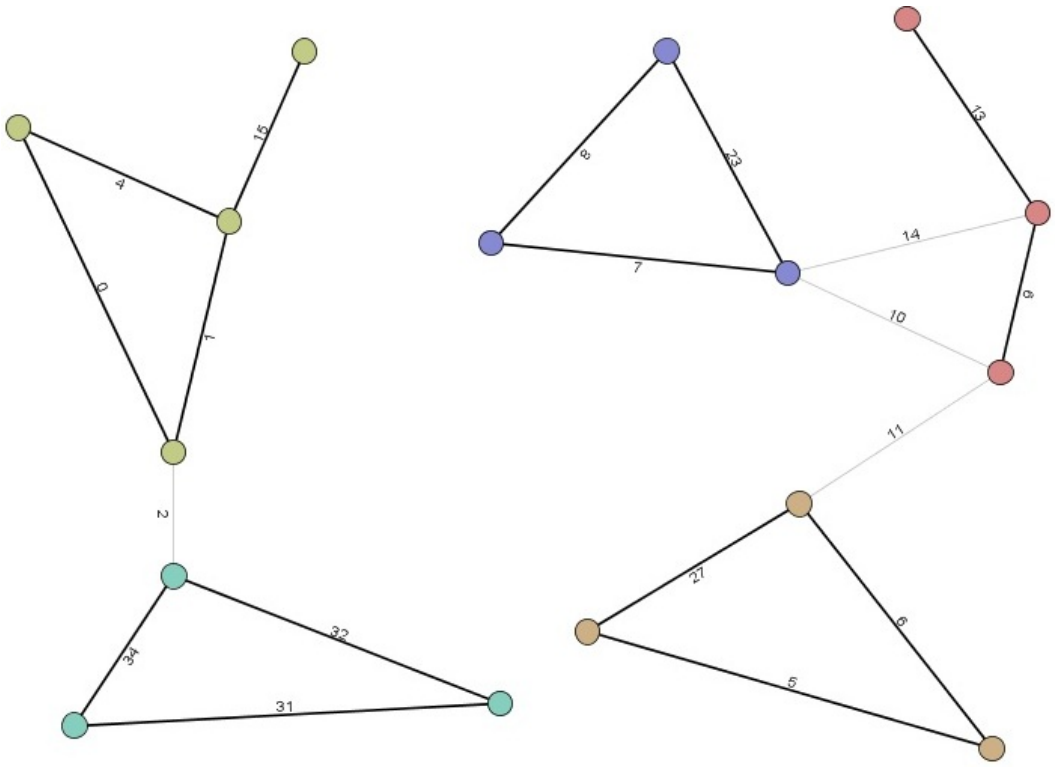


Figure B.2: After applying a clustering algorithm to the original network of Figure B.1. Four edges are removed as a result.

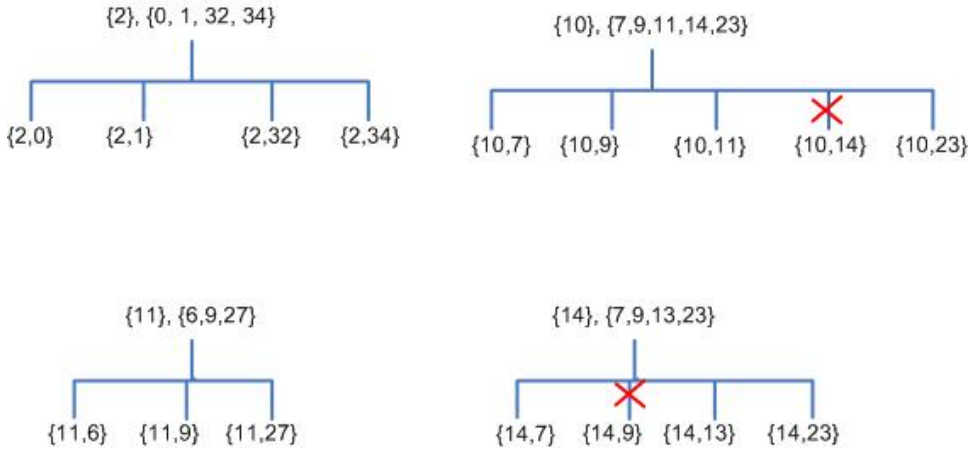


Figure B.3: The process of recovering missing subgraphs from removed edges of Figure B.2

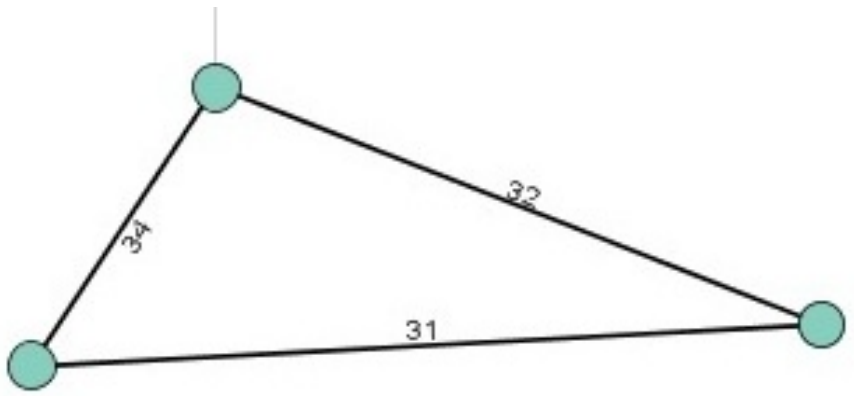


Figure B.4: After clustering, some clusters are isomorphic. For example, we obtain three clusters with this type of subgraph after clustering.

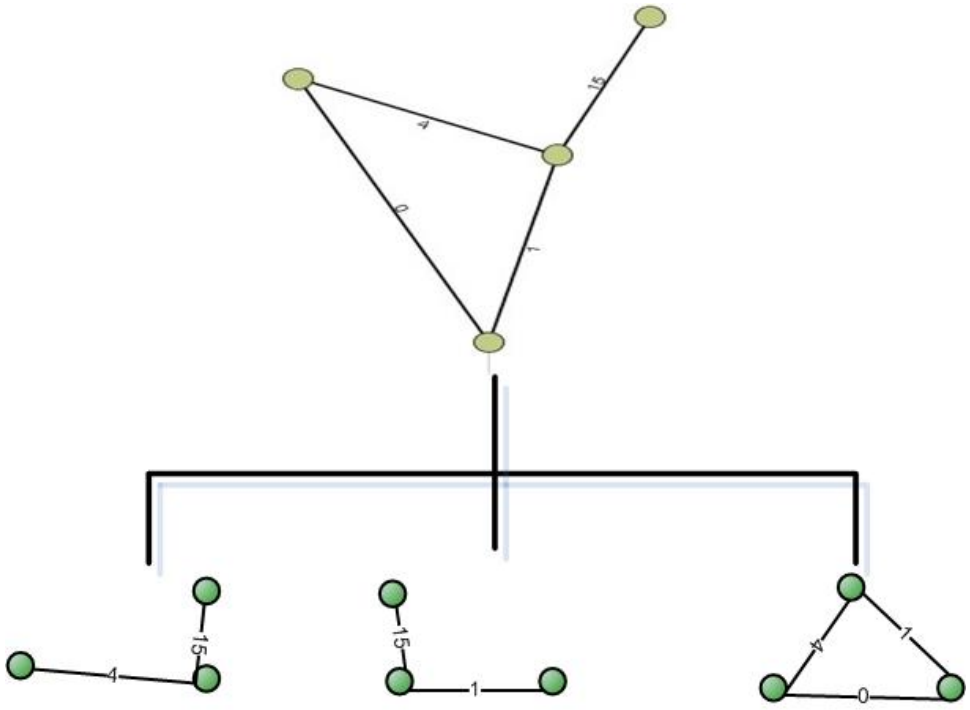


Figure B.5: 3-node subgraphs are enumerated using ESU [15] algorithm.