12-2013

Clustering and Classification of Multi-domain Proteins

Neethu Shah
University of Nebraska-Lincoln, nshah@cse.unl.edu

CLUSTERING AND CLASSIFICATION OF MULTI-DOMAIN PROTEINS

by

Neethu Shah

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Professors Stephen D. Scott and Etsuko N. Moriyama

Lincoln, Nebraska

December, 2013

CLUSTERING AND CLASSIFICATION OF MULTI-DOMAIN PROTEINS

Neethu Shah, M.S.

University of Nebraska, 2013

Advisers: Stephen D. Scott and Etsuko N. Moriyama

Rapid development of next-generation sequencing technology has led to an unprecedented growth in protein sequence data repositories over the last decade. Majority of these proteins lack structural and functional characterization. This necessitates design and development of fast, efficient, and sensitive computational tools and algorithms that can classify these proteins into functionally coherent groups.

Domains are fundamental units of protein structure and function. Multi-domain proteins are extremely complex as opposed to proteins that have single or no domains. They exhibit network-like complex evolutionary events such as domain shuffling, domain loss, and domain gain. These events therefore, cannot be represented in the conventional protein clustering algorithms like phylogenetic reconstruction and Markov clustering. In this thesis, a multi-domain protein classification system is developed primarily based on the domain composition of protein sequences. Using the principle of co-clustering (biclustering), both proteins and domains are simultaneously clustered, where each bicluster contains a subset of proteins and domains forming a complete bipartite graph. These clusters are then converted into a network of biclusters based on the domains shared between the clusters, thereby classifying the proteins into similar protein families.

We applied our biclustering network approach on a multi-domain protein family, Regulator of G-protein Signalling (RGS) proteins, where heterogeneous domain com-

position exists among subfamilies. Our approach showed mostly consistent clustering with the existing RGS subfamilies. The average maximum Jaccard Index scores for the clusters obtained by Markov Clustering and phylogenetic clustering methods against the biclusters were 0.64 and 0.60, respectively. Compared to other clustering methods, our approach uses auxiliary domain information of each protein, and therefore, generates more functionally coherent protein clusters and differentiates each protein subfamily from each other. Biclustered networks on complete nine proteomes showed that the number of multi-domain proteins included in connected biclusters rapidly increased with genome complexity, 48.5% in bacteria to 80% in eukaryotes.

Protein clustering and classification, incorporating such wealth of additonal domain information on protein networks has wide applications and would impact functional analysis and characterization of novel proteins.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Dr. Etsuko Moriyama for her continuous guidance and support for this thesis. I would also like to thank Dr. Stephen Scott for advising me throughout this thesis and also the M.S. program at CSE. I thank Dr. Ashok Samal for serving on my committee.

I thank Dr. Greg Sommerville and Dr. Jeffrey Mower for giving me the opportunity to work and collaborate with them during my graduate program at UNL.

I also appreciate the help of my lab members for making the work environment so much fun and interesting.

I take this opportunity to thank the administrative group of both Avery and Manter Hall. I specially thank Shelley Everett and Deb Heckens for all their help and support.

I thank my family, most of all my husband, for his love, support and encouragement without which this thesis would not have been possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recent advancements in high throughput sequencing have resulted in a massive accumulation of biological sequence data. Universal Protein Resource Knowledge Base (UniProtKB/TrEMBL), one of the fastest growing and globally maintained protein public databases, currently records 33,995,348 protein sequence entries comprised of 10,924,561,758 amino acids [5]. This database alone has shown a two-fold increase in the number of sequence entries just within the last two years. This exponential growth in biological databases poses direct challenges and therefore demands highly efficient and robust algorithms related to the major data mining components such as data integration, management, prediction and classification. Although the most important information for proteins is their functions, only a small portion of protein sequences available in such databases has been functionally characterized. Therefore, more accurate, sensitive, and efficient algorithms are necessary for the functional classification of protein sequences.

1.1 Proteins and their Domains

Proteins are polymers of amino acids that perform a wide variety of functions in living organisms. Besides acting as enzymes, hormones, and antibodies, proteins also perform the major regulatory functions in a cell. Some proteins contain recurring fragments that have distinct conserved structures and functions. These fragments within a protein are called "domains" and they act as fundamental units of protein structure and function [6]. They occur in single or multiple copies in a protein. Figure 1.1 shows five hypothetical proteins with domains in various combinations. Each protein maintains a unique combination and order of these domains. This unique domain arrangement can also be termed as the domain architecture of the protein [7]. Multi-domain proteins are complex in its structure, function, and evolution compared to single domain proteins. They constitute more than 65% of the protein databases such as CATH [8]. It has been shown that eukaryotes contain a larger proportion (approximately 70%) of multi-domain proteins in comparison to bacteria [9, 10]. High

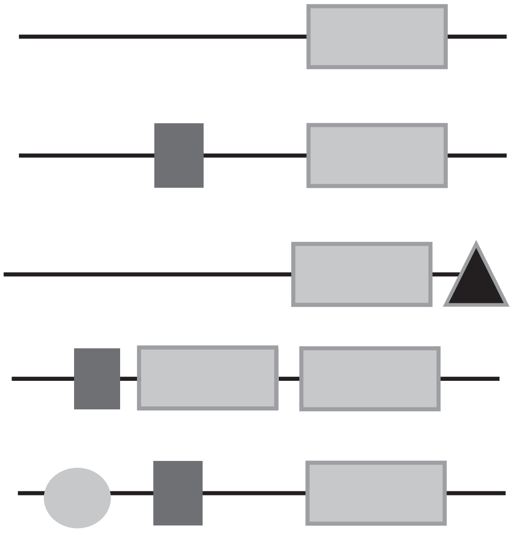


Figure 1.1: Five hypothetical proteins showing single to multiple copies of domains. Each shape—rectangle, square, triangle, and circle represents distinct domains.

proportions of multi-domain proteins in animals and plants account for their diverse and complex proteomes mediating functions such as protein-protein interaction, signal transduction, etc.

1.2 Multi-domain Protein Clustering

Clustering of protein sequences are critical since similar proteins perform related functions. Therefore, accurate and sensitive classification diverse proteins including those whose functions have not been identified can help predict their functions based on their similarities with known proteins [11]. However, prediction of their functions are dependent mainly on the degree of primary sequence similarity between them. Conventional clustering, such as phylogenetic clustering, is done based on information on sequence similarities from a single comparable region or domain of proteins. However, clustering and classification of multi-domain proteins are much more challenging as opposed to proteins with single or no domains. As illustrated in Figure 1.2, multi-domain proteins exhibit complex evolutionary events like domain shuffling, domain loss (deletion) or domain gain (insertion) [12]. These events are analogous to network properties and are not represented in conventional phylogenetic trees as phylogenetic reconstruction methods in general model evolutionary events that are passed via vertical descents only. As a result, information on the complex domain evolution such as horizontal transferring between proteins and duplications/deletions are lost or completely ignored. Therefore, a protein clustering method that can incorporate the similarity and difference in domain architectures is needed.

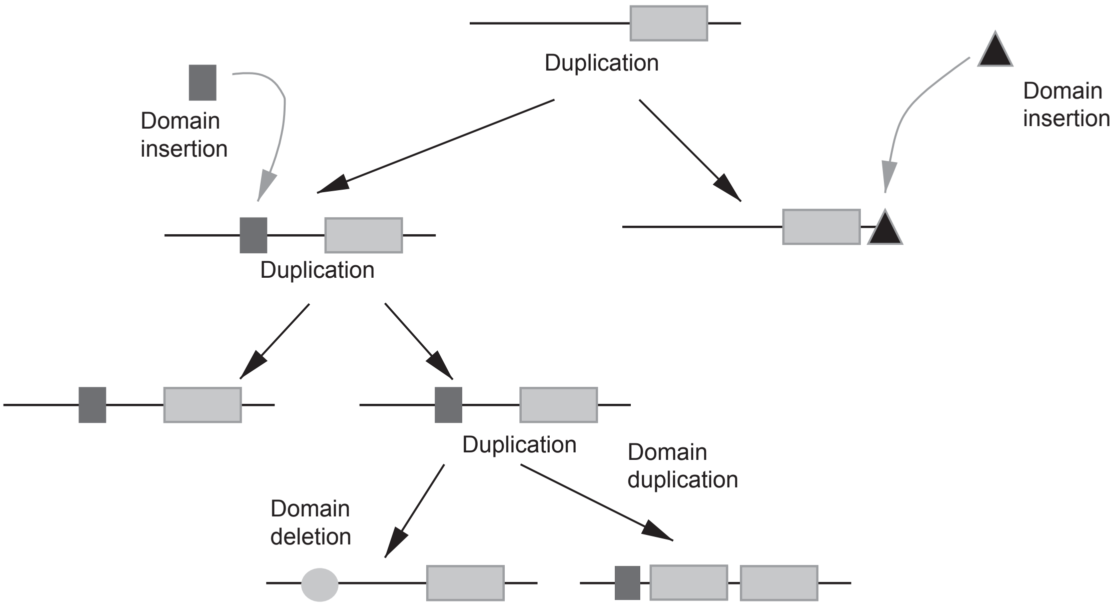


Figure 1.2: Evolutionary events of multi-domain proteins. It illustrates an example of a domain-containing protein evolving through various events such as domain insertion, domain deletion and duplication.

1.3 Objectives

The main objective of this thesis is to develop a classification system that would enable clustering and classification of multi-domain proteins. Such a method should be applicable to large-scale data at the multiple genome level. To achieve this goal an initial clustering of the entire set of protein sequences in terms of their domain composition is accomplished by the principle of bi-dimensional clustering. This method would enable us (a) to understand the complete evolutionary relationships between proteins including multi-domain proteins, represented by evolutionary networks of both proteins and domains at the same time, (b) to classify multi-domain proteins representing different protein families on a large and global scale, and (c) to compare such networks of all the protein families across multiple proteomes at varying levels of organismal complexity. To achieve the above mentioned research goals, this study has the following objectives:

1. to develop a protein-domain biclustering network for a given set of proteins,
2. to evaluate the proposed method against Markov and phylogenetic clustering methods, and
3. to compare multi-domain protein classes obtained from our method across different genomes of varying complexity.

Protein-domain biclustering network for a given set of protein sequences is developed by (a) identifying domains for each proteins and generating a protein-domain binary matrix (Section 4.2), (b) generating protein-domain biclusters using a biclustering algorithm Bimax [3], and (c) converting the set of clusters to a network of isolated and connected biclusters using the methodology described in Section 5.1.

This method was first applied on a multi-domain protein family—Regulator of G-protein Signalling (RGS) proteins, where heterogeneous domain composition exists among subfamilies, as shown in Section 5.2. Comparison of biclusters against Markov Clustering (MCL) [2] algorithm and maximum likelihood phylogenetic [13] method showed a high Jaccard Index scores for both these methods against the biclusters. These experiments are described in Section 5.2.1.

The final section of the results focusses on comparative analysis of protein-domain biclustered network across nine genomes including seven bacterial, one fruit fly, and one mouse genomes (5.3.2). Both bilcustering and MCL approaches were used to assess the clusters obtained at varying levels of E-value and also with varying domain prediction overlap thresholds — overlap and non-overlap domain predictions (Section 5.4).

Contributions to Bioinformatics and Computer Science Research Clustering and classification of proteins and domains as well as studies on domain organization using graph algorithms have been done previously. Some of these works are highlighted in Section 3. Nevertheless, these studies have focused so far on proteins and domains independently, and have not utilized the entire domain information when classifying multi-domain proteins. Only a handful of works have addressed protein clustering with respect to domain compositions. However, these works only focused on the representation of proteins and domains in a bipartite graph/biclusters (complete/incomplete). The works that proposed bipartite networks of proteins and domains were confined to a single species and their analyses solely focused on the organization and properties of the network. None of the works (described in Section 3) has attempted to establish any evolutionary or functional relationships between protein families based on protein-domain biclusters. In this study, we represent protein-domain graphs as a foundation to classify proteins into functionally coherent groups. A bicluster network approach is then developed (Section 5.1) to accomplish the classification.

The overall organization of the rest of the thesis is as follows. Chapter 2 describes the background information on sequence homology and similarity, protein classification, and protein function prediction. It also describes the principles of bi-dimensional clustering (Bimax) and Markov clustering of proteins. Chapter 3 analyzes related works on domain graphs and phylogenetic profile methods. Chapter 4 explains the methodologies in the construction of protein-domain similarity matrix, steps involved in developing a biclustering network, domain prediction algorithms, data sets used and parameters for cluster comparison. Results are presented in Chapter 5. The thesis concludes with Chapter 6, which includes overall discussion and future works.

Chapter 2

Background

This chapter describes: (a) background information on sequence homology, similarity, and protein functional classification (Section 2.1), (b) key features of profile Hidden Markov Model (pHMM) based sequence similarity detection (Section 2.2), (c) phylogeny of multi-domain proteins (Section 2.3), (d) Markov Clustering (MCL) algorithm [2] (Section 2.4), and (e) a bi-dimensional clustering algorithm Bimax [3], which is used for developing biclustered network of proteins and domains (Section 2.5).

2.1 Sequence Homology, Similarity, and Protein Functional Classification

Protein homologs are sequences that have arisen from a common ancestor. Sequence and structural similarities are used commonly to infer homology of protein sequences [14]. Identification of protein homologs has many practical applications. Protein homologs that are similar to each other are known to perform shared or related functions. Identifying diverse domains and protein sequences can help detect more remote homologs and therefore can provide information about the function, structure, and evolution of these proteins. This is the underlying principle followed by protein function prediction algorithms. Protein homologs that are highly conserved exhibit a higher degree of similarity and can be identified by sequence search algorithms like Basic Local Alignment Search Tool (BLAST)[15]. BLAST searches common "words" or k-tuples in the query and each database sequence. Using amino acid substitution matrices significant alignments of these words are estimated, which are extended to a larger stretch of sequence, until the High Scoring Pair (HSP) is found. However, as the degree of sequence similarity decreases more sensitive search strategies employing sequence profiles as in Position Specific Iterative BLAST (PSI-BLAST [16]) or pHMMs (*e.g.*, HMMER [17]) are used that would enhance the sensitivity of the searches made. In this study, we use pHMM-based search algorithm for domain identification. Therefore, the following section discusses the structure and mechanism of a pHMM.

2.2 Profile Hidden Markov Models and Domain Prediction

Sensitive search methods use information from a collection of similar proteins, such as, position specific scoring matrices [16] or multiple sequence alignment (MSA), rather than using a single sequence information. This composite information of multiple sequences is called a "profile". To construct such a profile of multiple sequence alignment, proteins that are similar to each other are aligned. For example, Figure 2.1 shows an example of multiple sequence alignment of seven protein sequences. A pHMM is derived from such a multiple sequence alignment [18]. The first step in constructing a pHMM is to define the states. The *match* state represents the residues

```
123456789
VGA--HAGE
V----NVDE
VEA--DVAG
VKG------
VYS--TYET
FNA--NIPK
IAGADNGAG
```

Figure 2.1: Multiple sequence alignment of seven protein sequences. First row of numbers represent the columns of the alignment. Alphabets represent amino acids and “-” symbols represent gaps in the alignment. In this example, columns 1-3 and 6-9 are “match” columns and coulmns 4 and 5 are “insert” columns. Modified from `http://www.cs.princeton.edu/~mona/Lecture/HMM1.pdf`

that are aligned to a residue rather than a gap. Portion of the sequence in the alignment that do not match with anything in the model is named as *insert* state. *Delete* state is the segment of the multiple sequence alignment that is not matched by any residue. The length of the pHMM determined by the number of match columns is estimated using several heuristics. One of the common heuristics is to include those columns that have at least half of the sequences as match columns. In this example, columns 1-3 and 6-9 are match columns making the length of the pHMM to be 7. Once the states are defined, the pHMM model structure can be built by calculating the transition and emission probabilities. Transition probability from state k to state l is given by the following equation.

$$a_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}}, \tag{2.1}$$

where, k and l are indices over the states, a_{kl} is transition probability and A_{kl} is the corresponding transition frequency. Transition probability $a_{M_1M_2}$ in this example is $\frac{6}{7}$. Similarly, $a_{M_1D_1} = \frac{1}{7}$ and $a_{M_1I_1} = \frac{0}{7}$, , where M_1, M_2 are match states, D_1 and I_1

are delete and insert states, respectively, as shown in Figure 2.2. Emission probability is given as,

$$e_k(a) = \frac{E_k(a)}{\sum_{a'} E_k(a')}, \tag{2.2}$$

where $E_k(a)$ is the frequency for the state k emitting the amino acid a. For example in the match state M_1 the probability of emitting amino acid V is given as, $e_{M_1V} = \frac{5}{7}$. To avoid zero probabilities a pseudo-count of 1 is used, and after accounting pseudo-count for each of the 20 amino acids, $e_{M_1V} = \frac{5+1}{7+20} = \frac{6}{27}$. Figure 2.2 shows the most likely path of this MSA.

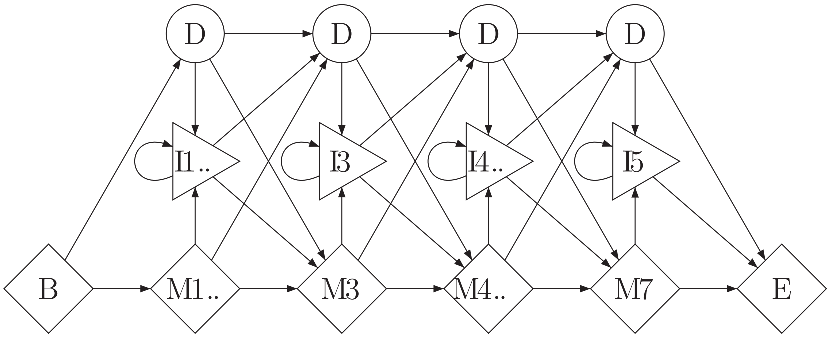


Figure 2.2: Structure of a pHMM. States include begin (B), match (squares), insert (triangles), delete (circles), insert (triangles) and end (E). Arrows show the transition probabilities between the states.

One of the main purposes of such a pHMM is to obtain significant match for a sequence against this profile and test its membership for the particular pHMM. This can be done by estimating the log-odds ratio of the probability of such a sequence x belonging to the HMM model, M given by $P(x \mid M)$ to that of the probability of the sequence to a random (null, N) model, $P(x \mid N)$.

$$S = \log \frac{P(x \mid M)}{P(x \mid N)}, \tag{2.3}$$

HMMER and Pfam HMMER [17], a sequence to profile search algorithm, is used in this thesis to identify domains for a given set of protein sequences. Pfam version 27.0 [19], a large collection of multiple sequence alignment and profile HMM libraries, was used as the underlying domain profile database for all domain searches. Pfam-A entry, used in this study is a high quality, manually curated 14,831 protein profiles database.

E-value as the Similarity Measure In this study, E-values are used as the scores of statistical significance showing a pair of sequence to be related or similar. The probability of getting the alignment score x or higher is obtained as,

$$(S \geq x) = 1 - exp(-Kmne^{-\lambda x}), \tag{2.4}$$

where K and λ are constants calculated from scoring matrix and amino acid composition (empirically calculated), and m and n are sequence lengths [20]. This is the Karlin Altshul statistics. E-value is the expected number of sequences in the data to have a score as high as or higher than the score S. In the case of pHMMs, Sean R. Eddy [21] made two conjectures about Viterbi and Forward scores in the case of full probabilistic models of local sequence alignment: (a) the Gumbel distribution of Viterbi scores has fixed $\lambda = logz$, where z is the base of the logarithm of the log-odds, and (b) the Forward scores is exponentially distributed with the same $\lambda = logz$. HMMER3 uses filters called "Viterbi filter" and "Forward filter" to evaluate profile-sequence comparisons. For the former, an optimal (maximum likelihood) gapped alignment score is calculated and the sequence is passed to the next step only if the score passes a set threshold. Forward filter calculates the likelihood by summing the entire alignment using the Forward algorithm and converts the score to a bit score.

A sequence is then evaluated based on this score.

2.3 Phylogeny of Multi-domain Proteins

Protein clustering algorithms generally perform a direct comparison of each sequence against every other sequence and establishes a “all-against-all” relationships between them (as in a similarity or distance matrix). These relationships are expressed in terms of the number of changes in amino acids. Given a set of multi-domain proteins, a phylogenetic tree is constructed based on the alignment of a common domain that exists in all of them. As an example, Figure 2.3 shows a phylogeny of 21 proteins that belong to the Regulator of G-protein Signaling (RGS) protein family [1]. While all these proteins share the common RGS domain, as shown in the figure (to the right), their domain architecture varies. Phylogenetic reconstruction of these protein sequences is, however, based on the alignment of only the common RGS domain sequences, and the rest of the sequence information is completely ignored. It illustrates that the phylogeny reconstructed only based on a small portion of protein sequences could easily fail to establish the complex evolutionary relationships among proteins that include mixed combinations of many domains. The evolutionary relationship thus established via a bifurcating phylogeny is often incomplete in terms of protein evolution.

2.4 Markov Clustering of Proteins

Markov clustering (MCL) algorithm is an unsupervised clustering algorithm for graphs or networks and is based on simulation of stochastic flow in graphs [22]. Protein clustering is one of the direct applications of MCL algorithm. TRIBE-MCL [2] uses the

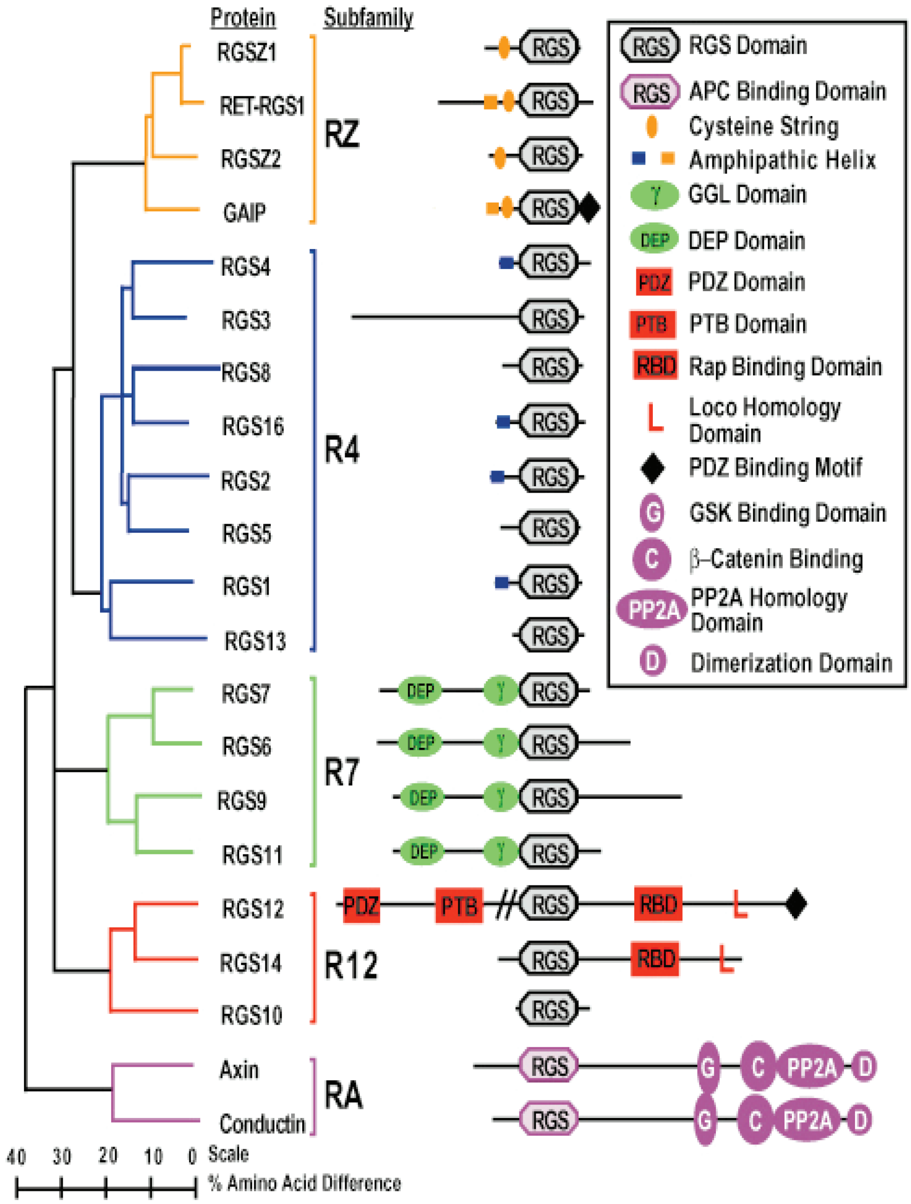
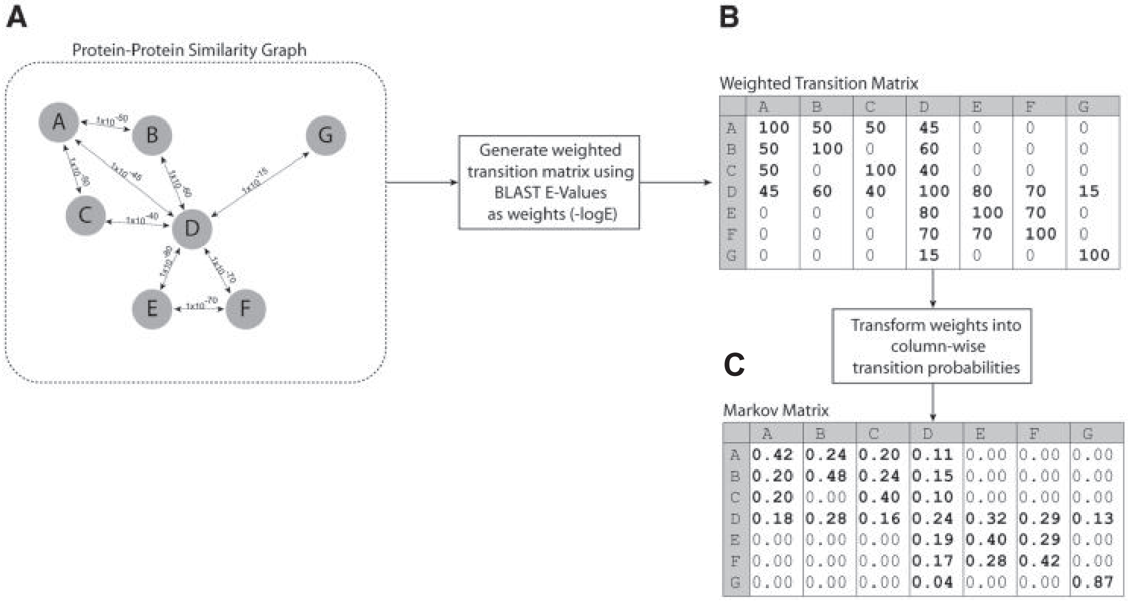


Figure 2.3: Phylogenetic relationship among RGS proteins. The domain architecture of each protein is illustrated on the right. [1]

MCL algorithm to classify protein sequences. It first performs all *vs.* all protein similarity search using BLASTp. A symmetric protein similarity matrix is generated


A
Protein-Protein Similarity Graph
A
B
G
C
D
E
F
$1x10^{-50}$
$1x10^{-50}$
$1x10^{-45}$
$1x10^{-60}$
$1x10^{-15}$
$1x10^{-40}$
$1x10^{-80}$
$1x10^{-70}$
$1x10^{-70}$
Generate weighted transition matrix using BLAST E-Values as weights (-logE)
B
Transform weights into column-wise transition probabilities
C

Weighted Transition Matrix

	A	B	C	D	E	F	G
A	100	50	50	45	0	0	0
B	50	100	0	60	0	0	0
C	50	0	100	40	0	0	0
D	45	60	40	100	80	70	15
E	0	0	0	80	100	70	0
F	0	0	0	70	70	100	0
G	0	0	0	15	0	0	100

Markov Matrix

	A	B	C	D	E	F	G
A	0.42	0.24	0.20	0.11	0.00	0.00	0.00
B	0.20	0.48	0.24	0.15	0.00	0.00	0.00
C	0.20	0.00	0.40	0.10	0.00	0.00	0.00
D	0.18	0.28	0.16	0.24	0.32	0.29	0.13
E	0.00	0.00	0.00	0.19	0.40	0.29	0.00
F	0.00	0.00	0.00	0.17	0.28	0.42	0.00
G	0.00	0.00	0.00	0.04	0.00	0.00	0.87

Figure 2.4: Overview of the TRIBE-MCL method. (A) An example of a protein-protein similarity graph for seven proteins (A–F). Circles represent proteins (nodes) and lines (edges) represent detected similarities based on BLASTp E-values. (B) The weighted transition matrix and (C) the derived column-wise transition probability matrix for the seven proteins. Taken from [2].

by removing the relations that violate symmetry in the matrix. This similarity matrix represents protein-protein similarity relationships, which can be considered as a weighted similarity graph as shown in Figure 2.4a. A weighted transition matrix is generated from BLAST E-values where a weight is calculated as $(-\log E)$ (Figure 2.4b). Then the values in the matrix are transformed into column-wise transition probabilities (Figure 2.4c).

The transition probability matrix is passed through the MCL algorithm to identify protein clusters as follows. The algorithm finds the cluster structure in a graph by a bootstrapping process. It first computes the probabilities of random walks through the sequence similarity graph. It uses two operators, inflation and expansion, to transform a set of probabilities into another. A stochastic column matrix is $M \in R^{k\times k}, M \geq 0$ and the sum of each column elements sum up to 1. Given the matrix M and a real number $r > 1$, after inflation, the resulting matrix is written as $\Gamma_r M$, where Γ_r is the inflation operator with power coefficient r. $\Gamma_r : R^{k\times k} \rightarrow R^{k\times k}$ is defined by,

$$(\Gamma_r M)_{pq} = \frac{(M_{pq})^r}{\sum_{i=1}^{k} (M_{iq})^r}. \tag{2.5}$$

For values of $r > 1$, inflation changes the probabilities for a particular group of random walks by choosing more probable walks over less probable walks [2]. Given a start node and a destination node, expansion represents the path lengths of the random walks. Expansion scatters the stochastic flow within the clusters, where as inflation eliminates flow between the clusters. Iteration of inflation and expansion separates the graph into segments. An equilibrium state is reached when no change is observed in the matrix after a series of expansion and inflation.

This method has been shown to detect protein families accurately on a large scale dataset. A large proportion of protein families from the human genome was classified using this method. However, both this method and the phylogenetic clustering approaches fail to include the complete domain information of a protein. When phylogeny uses the common domain information, MCL uses the most significant region between the proteins. These methods therefore, ignores the auxiliary domain information of all the multi-domain proteins in the respective clusters.

2.5 Bidimensional Clustering (Biclustering)

This method was first introduced by the name “direct clustering” where voting data was clustered to the states that voted similar candidates with respect to the years [23]. The concept of biclustering in biological samples was first introduced by Cheng and Church where they used simultaneous row-column clustering to cluster gene expression data [24] to isolate genes (rows) that are expressed in certain similar conditions or samples (columns). The name "biclustering" is also interchangeably used with co-clustering, bidimensional clustering, subspace clustering, etc [25]. Even though several biclustering algorithms have been developed for most predominantly clustering gene expression data, this concept is also widely used in the field of text mining, web mining, etc [26].

2.5.1 Definitions

Let us consider two variable sets X, Y where $X = \{x_1, x_2, \ldots, x_n\}$, and $Y = \{y_1, y_2, \ldots, y_m\}$. Given these two sets, the problem of biclustering is formally defined as finding the set of biclusters $B = \{B_1, B_2, \ldots, B_k\}$ such that $B_l = (X_l, Y_l)$ where $X_l \subseteq X$ and $Y_l \subseteq Y$ and $l \in \{1, 2, \ldots, k\}$. The variables X and Y could represent different variables such as genes-conditions, texts-words, webpages-contents, and proteins-domains.

Bicluster data as a bidimensional matrix. Let us consider a general case of two-dimensional matrix A with n rows and m columns. The variable sets X and Y could represent the respective rows and columns of the matrix and the cells bear a real value attribute type. That is, let the matrix A has the set of rows to be X where $X = \{x_1, x_2, \ldots, x_n\}$, and set of columns to be Y where $Y = \{y_1, y_2, \ldots, y_m\}$.
A bicluster $B_l = (X_l, Y_l)$ is a submatrix of A such that, $\forall a_{ij} \in A_{X_l Y_l}$, $i \in X_l$ and $j \in Y_l$. A biclustering problem can be formally defined as the problem of finding a set of sub-matrices $\{(X_1, Y_1), (X_2, Y_2), \ldots, (X_k, Y_k)\}$ of the matrix $A = (X, Y)$, where $X_i \subseteq X$ and $Y_i \subseteq Y; \forall i \in \{i, \ldots, k\}$, such that every submatrix meets a given pattern or homogeneity criterion (*e.g.*, a certain significant E-value threshold for a

protein-domain matrix described in Section 4.2).

Bicluster data as a bipartite graph A bipartite graph is a graph $G = (U \cup V, E)$ with two disjoint vertex sets U and V such that every edge in E connects a vertex from U to V. A bilcluster data can be transformed into a bipartite graph $G^{'} = (X \cup Y, E^{'})$ (Figure 2.5) where the variable sets X and Y (the rows and columns in a matrix) form the vertex sets of the bipartite graph $G^{'}$, and the edges are the real value attributes of the matrix cells.

A bicluster $B_l = (X_l, Y_l)$ is a subgraph $g = (X_l \cup Y_l, W_l)$ of $G^{'}$ such that $\forall w_{ij} \in W_l$, $W_l \subseteq E^{'}$, $i \in X_l$ and $j \in Y_l$. A biclustering problem thus formally defined as the problem of finding a set of subgraphs $\{(X_1 \cup Y_1, W_1), (X_2 \cup Y_2, W_2), \ldots, (X_k \cup Y_k, W_k)\}$ of the bipartite graph $G^{'} = (X \cup Y, E^{'})$, where $X_i \subseteq X \wedge Y_i \subseteq Y$ and $W_i \subseteq E^{'}; \forall i \in \{i, \ldots, k\}$, such that every subgraph meets a given pattern or homogeneity criterion.

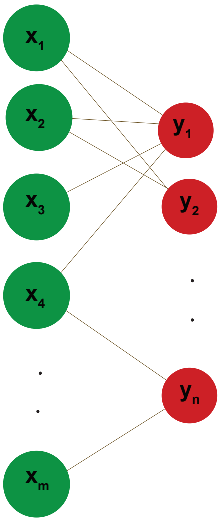


Figure 2.5: Bicluster data represented as a bipartite graph. A bipartite graph $G = (X \cup Y, E^{'})$, where two variable sets X and Y are shown with the colors green and red, respectively.

2.5.2 Biclustering Approaches.

Although the run time complexity of biclustering problems varies depending on how the problem is formulated, almost all of the biclustering problems are NP-complete [25]. For a given binary matrix M, with $m_{ij} \in \{0, 1\}$, a bicluster here is equivalent to a biclique or complete bipartite graph. For instance, the problem of finding a maximum sized biclique can be transformed to the problem of finding maximum edge biclique in a bipartitie graph. This problem is a known NP-complete problem [27]. A few heuristic approaches in biclustering two-dimensional data are described below.

The method developed by Cheng and Church, one of the earliest, clustered gene expression data using a brute force method. The key idea was to find the largest submatrix with the lowest mean squared residue [24]. There are several modifications of this algorithm. One of them is XMOTIF. In this method each bicluster represents a conserved gene expression motif, which contains a subset of genes whose expression patterns are simultaneously conserved for a subset of samples (*e.g.*, from different tissues) [28]. If genes and samples are represented by rows and columns in a matrix, respectively, this is equivalent to a set of rows that shares a specific range of values for a specific set of columns. Another method (Samba), developed by Tanay et al [29], looks for heavy subgraphs in a bipartite graph. It uses the idea of maximum bounded biclique to find the maximum bounded bipartite subgraphs. Given a bipartite graph, G with two sets of vertices representing genes and conditions, it finds the maximum weight subgraphs (not necessarily complete) of G, where the vertices from the gene side maintains a certain vertex degree. The Order-Preserving Submatrix (OPSM) algorithm [30] generates submatrices where, each submatrix is preserved in terms of its order such that for the set of columns the sequence of values of the rows is strictly increasing. In contrast to the values of a cluster in XMOTIF where the values of the rows are within a specific range, in this method the values of the rows with respect to the column is strictly positively correlated. The common feature in all of these algorithms is that clustering is based on the real valued attribute type that relates the two sets of variables in a bicluster. Unlike all of the above mentioned approaches, Bimax clusters a binary model into submatrices where members of both the row and column sets are connected to each other [3]. Detailed description of this algorithm is given in Section 2.5.3

2.5.3 Binary Inclusion-Maximal Biclustering (Bimax)

Inclusion-maximal biclusters are defined as those that are not strictly contained in any other biclusters. Two biclusters B_i and B_j are inclusion maximal if,

$$B_i = (X_i, Y_i) \nsubseteq B_j = (X_j, Y_j).$$

Biclusters here are also completely connected bipartite graphs, where every vertex of the first set X_i (*e.g.*, proteins) are connected to the second Y_i (*e.g.*, domains). For example, from a bipartite graph shown in Figure 2.6a inclusion-maximal biclusters shown in Figure 2.6b can be derived. Bimax is a heuristic algorithm that finds all the inclusion-maximal biclusters from a two-dimensional matrix. It works on a binary model by the divide and conquer strategy [3]. The algorithm works as follows. It first chooses a row as the template and partitions the column set C into C_U and C_V (Figure 2.7, left). The heuristic is to choose a row i such that the statement, $0 < \sum_{j \in C} e_{ij} <\mid C \mid$ holds true. In the example shown in Figure 2.7, the first row is chosen as the template. C_U contains the columns in which the template has 1s (dark cells in Figure 2.7). $C_V = C - C_U$. Then, it sorts rows into three sets—G_U, G_W, and G_V (Figure 2.7, right). G_U is the set of rows that have 1s in C_U only, G_V in C_V only,

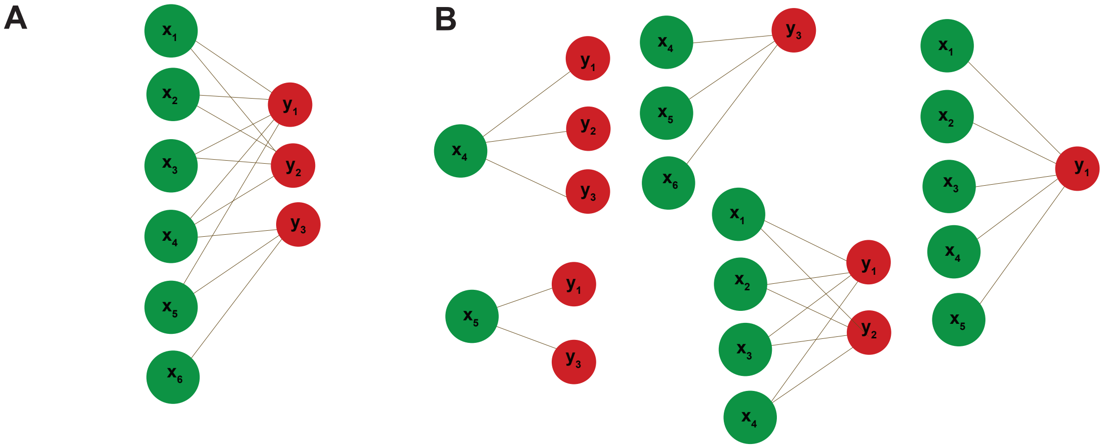


Figure 2.6: Inclusion-maximal biclusters. (A) A bipartite graph with two sets of vertices $\{x_1, \ldots, x_6\}$ (green) and $\{y_1, y_2, y_3\}$ (red). (B) six inclusion maximal biclusters are derived from the bipartite graph shown in A.

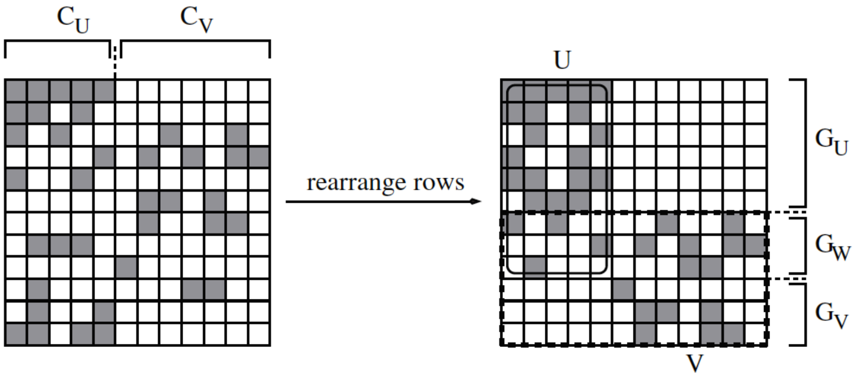


Figure 2.7: Bimax algorithm. Submatrices U and V are marked as boxes with solid and dashed lines, respectively, within the matrix to the right. Taken from [3]

and G_W where 1s are present in both C_U and C_V. The key idea is to partition the matrix into three submatrices, $U = (G_U \cup G_W, C_U)$, $V = (G_W \cup G_V, C_U \cup C_V)$, and those that contain only 0-cells. As is the case in Figure 2.7, if G_W is not empty, the matrix V contains parts of biclusters that are in U. Regarding that the algorithm finds the biclusters that are inclusion-maximal, it considers biclusters in V that extends over C_V. This process is then applied recursively on the matrices U and V until all the cells in the submatrix are 1s.

Biclustering in biology. Biclustering algorithms in biology are most commonly used to cluster gene expression data. As described in Section 2.5.2, numerous algorithms were developed just for the application of clustering genes with respect to biological samples [24, 30, 28, 29, 3]. However, biclustering on protein-domain data has been studied only in limited cases. A study on yeast proteins using bipartite network model of proteins and domains were used to identify co-occurring domain sets [31]. Later, it were demonstrated that unlike domain graphs that follow a scale free distribution, protein-domain networks have much more complex patterns. Using the human proteome, they showed that when the degree distribution for the number of domains shared by certain k proteins has a power law distribution, the degree distribution for the number of proteins composed of k types of domains follows an exponential decay [32]. However, these works were limited to the study of network properties and co-occurring domain sets in protein-domain networks and were not on clustering proteins into similar protein families.

Chapter 3

Related Works

This chapter describes: a few fundamental concepts and current research findings on (a) protein-domain clustering including phylogenetic profile method (Section 3.1), (b) domain clustering (Section 3.2), and (c) protein-domain networks (Section 3.3).

Several graph-clustering and network-based approaches have been developed to classify and predict functions of complex multi-domain proteins. Domain organization, co-occurrence, and orders have been intensively studied using domain graphs [33, 34, 35]. Besides evolutionary relationships of protein families, protein and domain co-occurrence networks contribute to functional classification of proteins [36, 37, 38]. Even though protein and domain graphs are studied intensively, complete association of the proteins to domains and also their relations to their functions remain to be explored. In the phylogenetic profile method, GDDA-BLAST [4], domain information of a protein is encoded as a profile. It is a multiple sequence alignment free method in classifying proteins. This chapter summarizes few key works on domain graphs and functional prediction of proteins.

3.1 Phylogenetic Profile Methods

Gestalt Domain Detection Algorithm-Basic Local Alignment Tool (GDDA-BLAST) developed by Chang et al [4] constructs evolutionary relationships among highly divergent protein sequences including multi-domain proteins. This method is based on phylogenetic profiles constructed for each of protein sequences. It is independent of multiple sequence alignment, which is required for conventional phylogeny reconstruction methods (Section 2.3).

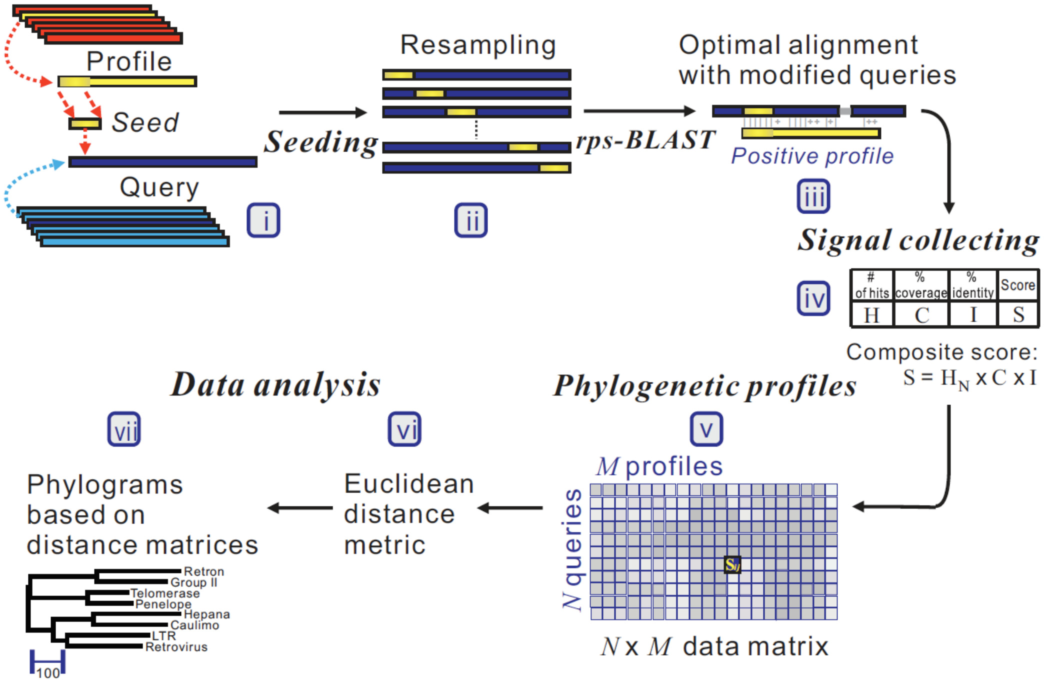


Figure 3.1: Workflow of the phylogenetic profile method with GDDA-BLAST. Taken from [4]

As illustrated in the Figure 3.1, the method compares each of query sequences against the domain profile data set. The profile data can be obtained from, for example, from National Center for Biotechnology Information Conserved Domain

Database 24,280 domain profiles. Each query sequence is first modified as follows. Seed sequences are generated from each profile by taking N- and C-terminal portions (*e.g.*, 3-50%) of one of the sequences. These seed sequences are inserted between each amino acid position of a query producing modified query sequences as many as the total number of amino acid in the sequence. Then the optimal pairwise alignment is generated using reverse PSI-BLAST [39] between each of the modified query sequence and each profile. For each comparison between a profile and a query, a composite score is defined as the product of mean percent coverage, mean percent identity, and the normalized hit number. All scores between queries and domain profiles are represented in an N x M matrix, where N is the total number of queries and M is the total number of profiles. This matrix is then converted into an Euclidean distance matrix and a phylogenetic tree inference method is used to construct the phylogeny.

Comparison of the GDDA-BLAST based phylogeny to a phylogeny constructed by using a regular method based on the multiple sequence alignment of the common domain in all the 88 sequences aligned using Dialign [40] showed a high similarity in their topology. The GDDA-BLAST based method could establish relationships for extremely diverse sequences that were used in the study. However, GDDA-BLAST's scalability is questionable as it is tested on a very small subset of 88 sequences.

3.2 Domain Networks

Domain clusters or topology of domain networks generated from a specific genome or from large scale databases have been studied comprehensively in the past (*e.g*, [41, 33]).

A domain graph $G_d = (V_d, E_d)$ is defined as an undirected graph that consists of a

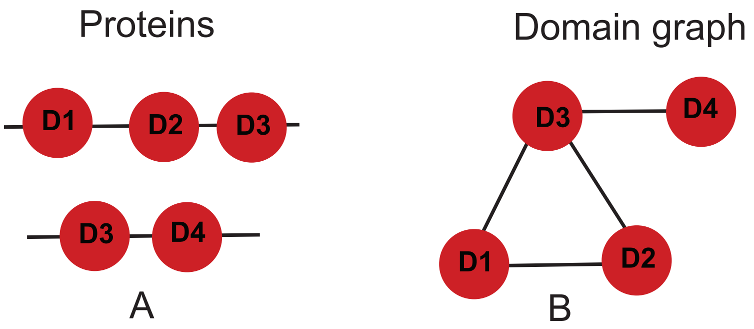


Figure 3.2: Domain graph. Two proteins with their domain organizations (A). The domain graph is generated with domains (vertices) in each protein forming a clique (B). The five domains from the two proteins form a graph with four vertices.

vertex set V_d representing all domains in a given set of proteins and a set of edges E_d. Two vertices are linked with an edge if the two domains are both present in at least one protein 3.2. The degree k of a vertex is the number of other vertices a vertex is linked to. Wutchy [41] demonstrated that the connectivity distribution $P(k)$ of nodes decays as a power-law given by,

$$P(k) \sim k^{-\gamma}. \tag{3.1}$$

Domain graphs were further used to compare and study domain organization of proteins in various organisms [33]. They analyzed structure, connectivity, and modularity of domain graphs across several genomes. Some of the key findings from this work were that the number of domains, the number of domain combinations, and the size of the largest component increase with the complexity of the organisms. Wutchy and Alamas [34] identified evolutionary cores of domain graphs by developing a k-core decomposition method that isolated globally central (highly connected domains in the central cores) from the locally central (highly connected domains in the peripheral cores) domains through an iterative method [34]. The k-core of a graph is the largest subgraph where every node has at least k links. This method recursively prunes all the nodes with degree less than k, for a given k. This study showed that the innermost k-core is not populated by the largest hubs, which indicated that a hub alone does not imply a central placement in the network.

Protein domain organization was also later studied by adding directionality to the graphs to represent the specific order by which the domains exist in proteins [35]. Evaluation of these directed network graphs was performed by comparing the observed values of global network properties to those expected at random. Random graphs were generated to emulate the scale-free behavior of observed graphs. The algorithm developed maintains the degree distribution of the nodes but removed all the original edges. Through a series of iteration, the algorithm randomly selects a node pair each from an in and out degree list, and defines a new edge and there by completes the new graph. One of the novel findings from this work was the presence of domain pairs to exist in both forward and reverse orders in proteins more often than random in contrast to what previous studies had shown.

3.3 Protein-Domain Networks and Protein Functions

Protein-domain networks, in contrast to protein networks or domain co-occurrence networks, provide comprehensive representations of both proteins and domains simultaneously and also their associations. These networks provide information such as co-occurring domain sets, domain distribution, and domains shared across protein families. These networks can further be used to classify proteins into functionally coherent groups, as domains are functional units of proteins. Unlike conventional protein or domain clustering methods, these networks can provide insight into complex evolutionary events such as domain recombination, domain shuffling, domain gain, and loss. Literature review shows that only a limited number of works have been done to study such networks.

A study on yeast proteins using a bipartite network model of proteins and domains was conducted to identify co-occurring domain sets [31]. Later, it was demonstrated that unlike domain graphs that follow a scale free distribution, protein-domain networks have much more complex pattern [32]. Using the human proteome, they showed that, when the degree distribution for the number of domains shared by certain k proteins has a power law distribution, the degree distribution for the number of proteins composed of k types of domains follows an exponential decay [32]. However, these works are limited to the study of network properties and co-occurring domain sets in protein-domain networks. A network of both proteins and domains connected with respect to the shared domain types is the potential novelty that we would accomplish in our work.

Chapter 4

Methods

This chapter describes all the methodologies used in this thesis in developing a protein-domain network for a given set of multi-domain proteins. It describes: (a) an overview of the protein-domain biclustering and the subsequent network approach developed that clusters proteins into groups of similar protein families (Section 4.1), (b) the structure of protein-domain binary matrix (Section 4.2), (c) domain identification and overlapping and non-overlapping domain predictions (Section4.4), (d) the data sets used (Section 4.5), and (e) the evaluation of the methods (Section 4.7).

4.1 Domain Content Based Clustering of Proteins – the Workflow

Figure 4.1 shows the complete workflow of the protein-domain biclustering and network construction. It starts with the domain identification for a set of proteins using profile HMM search using the HMMER (version v3.1) programs against the domain database Pfam (version 27) (A). A protein-domain binary matrix is constructed based on a given E-value threshold (B). E-value is a score of statistical significance which determines whether two sequences are significantly similar to each other or not (Section 4.3). Protein-domain inclusion maximal bicliques are identified using Bimax (C). After these steps, biclusters are converted into a network (D) with biclusters (nodes) connected by edges with the number of their shared domains (E). Steps (D) and (E) are described in Section 5.1.

4.2 Protein-Domain Binary Matrix and Bimax

We consider a protein-domain matrix M, with r rows (proteins) and c columns (domains) as shown in Table 4.1. Let the sets of rows and columns be $P = \{p_1, p_2, \cdots, p_r\}$ and $D = \{d_1, d_2, \cdots, d_c\}$, respectively. These two sets are analogous to the variable sets X and Y defined in Section 2.5.1. Every cell m_{ij} holds a value, *e.g.,* similarity measure such as percentage identity or an E-value obtained from domain prediction algorithms such as HMMER. Such a two-dimensional matrix becomes the basis of generating a set of biclusters $B = \{B_1, \ldots, B_k\}$. In the context of proteins and domains, a bicluster $B_n = (P_n, D_n)$ (defined in Section 2.5.1) is a sub-matrix $M_{P_n D_n}$ of M such that $P_n \subseteq P$ and $D_n \subseteq D$ and all the cell values m_{ij} maintain a pattern of similarity, for example, having E-values all within a certain threshold. In order to apply Bimax, all cells values in the matrix are binarized to 0 or 1 with respect to the chosen E-value threshold. For instance, all the cells with E-values greater

Table 4.1: Matrix representation of proteins and domains.

	Domain 1	...	Domain j	...	Domain c
Protein 1	m_{11}	...	m_{1j}	...	m_{1c}
Protein ...	...	...	...	...	...
Protein i	m_{i1}	...	m_{ij}	...	m_{ic}
Protein ...	...	...	...	...	...
Protein r	m_{r1}	...	m_{rj}	...	m_{rc}

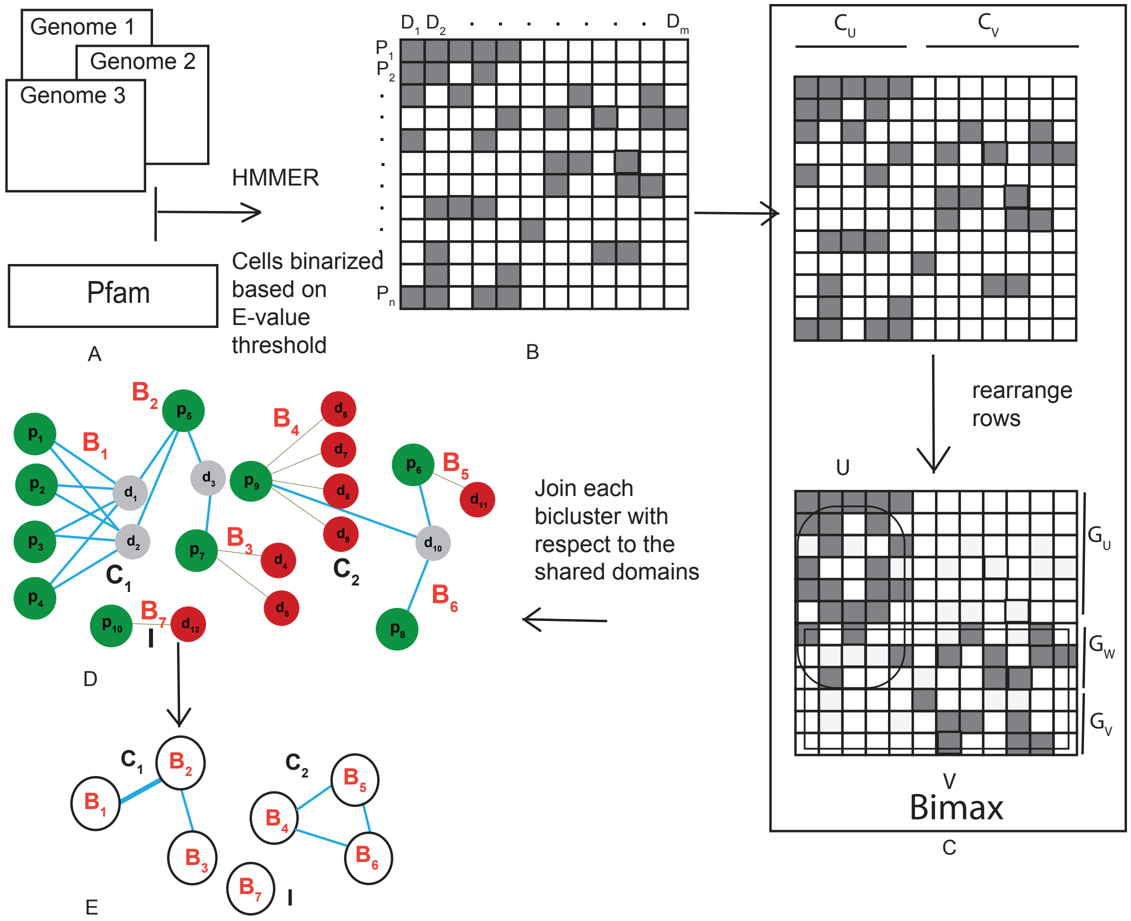


Figure 4.1: Workflow of protein-domain biclustering and network construction. (A) Domain identification from proteins using HMMER3 against the Pfam Database. (B) Domain predictions represented in a matrix with rows as proteins and domains as columns. Cells contain E-values for domain identification from each protein. Predictions below a defined E-value threshold are binarized (dark cells are 1 and light cells are 0). (C) Bimax algorithm is run on the matrix. (D) Network of multi-domain proteins containing connected (*e.g.*, C_1, C_2) and isolated (*e.g*, I) biclusters (B_1-B_7). Shared domains are colored grey and the edges that join the shared domains connecting biclusters is colored blue. (E) Network reconstruction where each bicluster forms the vertex and the edge weight is proportional to the number of shared domains between them.

than 1.0 (low similarity) are set to 0 and the rest (high similarity) to 1. This binary matrix can be used as the input of Bimax. Detailed description of the algorithm is given in Section 2.5.3. The C library implementing Bimax was downloaded from `http://people.ee.ethz.ch/~sop/bimax/`.

4.3 E-value as the Similarity Measure

E-value is defined as the expected number of of sequences in the sequence data to have a score as high as or higher than a particular alignment score. It is otherwise the score of statistical significance which shows a pair of sequence to be related or similar. Protein-domain biclusters differ in number, composition and structure with respect to varying levels of E-value. These differences are compared at three levels of E-value: 10, 1, and 0.001, respectively.

4.4 Overlapping and Non-overlapping Domain Predictions

Depending on how domains are modeled, it is possible to have predicted domain regions to be overlapped within a protein sequence. To examine the effect of this problem, we analyzed protein-domain biclusters with and without allowing overlapping domain predictions as follows: (1) including all domain predictions identified within a given E-value threshold regardless of overlapped or not, (2) excluding domain predictions whose overlaps are longer than 5% of the protein length (if the protein length is 1000 amino acids, only 50 amino acids or shorter overlaps are allowed; for domains that have longer overlaps, only those with the highest E-values are kept), and (3) including only strictly non-overlapping domains. Figure 4.2A illustrates when all

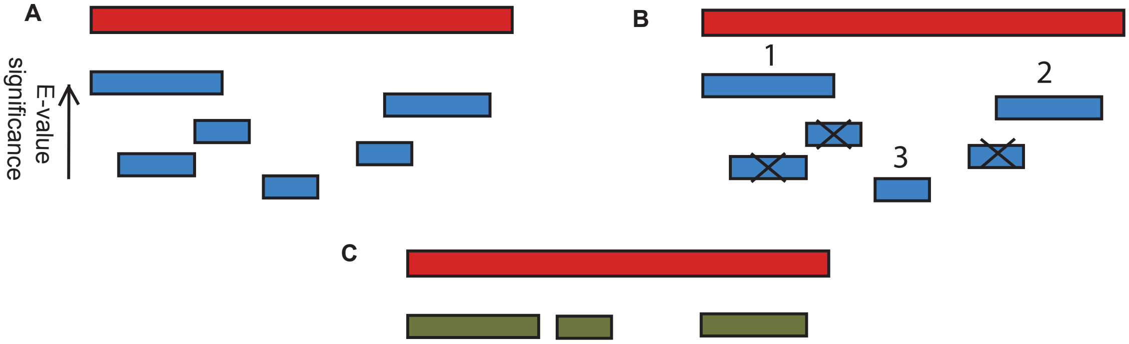


Figure 4.2: Overlapping domain predictions. (A) For a protein sequence (red bar), domains (blue bars) are predicted by HMMER. Some protein regions are predicted to have more than one domain. They are called "overlapped" . (B) In order to choose non-overlapping domains, first "domain 1" that has the lowest E-value is chosen and the two overlapping domains with the domain 1 are discarded (marked with "x"). Next the domain that has the second lowest E-value (domain 2) is chosen and the process is repeated. (C) Finally, three non-overlapping domains (green bars) are chosen for this protein.

predicted domains above the E-value threshold (blue bars) are included. To choose a set of non-overlapping domains first the domain with the lowest E-value (domain 1 in Figure 4.2B) is chosen and other overlapped domains are removed. Next the domain with the second lowest E-value (domain 2 in Figure 4.2B) is chosen and the process is repeated until no domain remains. Finally, as illustrated in Figure 4.2C, three non-overlapping domains (green bars) are chosen.

4.5 Data Sets used in this study

Two types of protein data sets were used in this study. The data set of 66 proteins from the Regulator of G-protein Signalling (RGS) family proteins from the mouse (*Mus musculus*) genome and the respective domain predictions are listed in Table A.1. These proteins contain at least one RGS (Pfam ID: PF00615) or RGS-like domain

(Pfam ID: PF09128). Fifty five non-overlapping domains were identified from the 66 of mouse RGS proteins (Table A.1).

Nine complete protein sets were also examined in this study. They were obtained from seven bacterial and two eukaryotic (*Drosophila melanogaster* and *Mus musculus*) genomes. Bacterial genomes were downloaded from National Center for Biotechnology Information (`www.ncbi.nlm.nih.gov/`). The complete protein set of the *Drosophila melanogaster* genome (version r5.52) was downloaded from the FlyBase database (`flybase.org`). The complete mouse genome (Taxonomy ID 10090) was downloaded from National Center for Biotechnology Information (`www.ncbi.nlm.nih.gov/`. From each data set, domains were identified using HMMER and PFAM with different inclusion strategies for overlapping domains. These data sets are listed in the Table 4.2.

Table 4.2: The nine complete protein data sets used in this study.

Genomes	**Total proteins**	#Overlap domains^a	#Overlap domains (5%)^b	#Non-overlap domains^c
Bacillus subtilis strain 168^p	4251	5095	3098	2822
Staphylococcus aureus strain COL^p	2680	3898	2306	2097
Staphylococcus epidermidis strain FRI909^p	2268	3696	2086	1905
Streptococcus pyogenes strain MGAS10270^p	1964	2993	1784	1645
Escherichia coli strain ATCC 33849^p	4588	4720	3128	2909
Yersinia pestis strain D106004^p	3642	4301	2833	2636
Treponema pallidum strain SS14^p	1028	1767	1088	983
Drosophila melanogaster^e	29,217	8277	6545	5931
Mus musculus^e	29,281	9806	7654	6885

^{a}: The domain data set includes all overlap domains below the E-value threshold of 1. ^{b}: The domains below the E-value threshold of 1 and with less than 5% of overlap are included. ^{c}: Only non-overlap domains below the E-value threshold is 1 are included. ^{p}: prokaryotes. ^{e}: eukaryotes.

4.6 Evaluation of Protein-Domain Clustering

4.6.1 Phylogenetic Clustering

As described in Section 2.3, reconstructing phylogeny requires a multiple sequence alignment generated from the domain sequences shared among all proteins. Therefore, this method can be used only for the RGS protein set in this study. A multiple sequence alignment of RGS protein sequences was generated using MAFFT (version v7.050b [42]) using the L-INS-i algorithm with the default parameters. The maximum-likelihood phylogeny was reconstructed as implemented in PHYML (version v3.0 [13]) using the following options:

```
phyml -i rgs.ph -d aa -m LG -a e -b 1000,
```

where "rgs.ph" is the multiple sequence alignment of RGS proteins. The option "-m LG" uses the LG amino-acid substitution model, "-a e" specifies the gamma distribution shape parameter with the maximum-likelihood estimate, and "-b 1000" specifies the bootstrap analysis with 1000 pseudoreplicates. We used a bootstrap of 70% to define the clusters of RGS sequences. For example, the sequences that has a bootstrap support of 70% or higher belong to a cluster and the rest form clusters, each with a single protein. A total of 20 clusters were obtained from this process as shown in Figure 5.5.

4.6.2 Markov Clustering

The Markov Clustering (MCL) algorithm is used for comparing the protein clusters with the biclusters. The library for MCL algorithm is downloaded from the webpage `www.micans.org/mcl`. The details of TRIBE-MCL were described in Section 2.4. It includes the following steps: (a) for a set of proteins, a hit table is generated using the `blastp` program, (b) the program `mcxdeblast` is used to parse and construct the all-against-all similarity matrix based on an E-value threshold of 1.0, (c) the program `mcxassemble` further generates a probability matrix from the blast matrix and also checks for the symmetry for each cell in the matrix. These steps are required for the final clustering, and (d) the program `mcl` is then used to cluster the matrix.

4.7 Evaluation Metric for Cluster Comparison

We used the maximum average Jaccard Index [3] to assess the performance of biclustering compared against the protein clusters generated by MCL as well as by phylogenetic clustering. Given two sets of protein clusters, B from Bimax and M from an alternative method, the average maximum Jaccard Index against the alternative method is given by,

$$S(B, M) = \frac{1}{\mid B \mid} \sum_{B_1 \in B} max_{(M_1) \in M} \frac{\mid B_1 \cap M_1 \mid}{\mid B_1 \cup M_1 \mid}. \tag{4.1}$$

Chapter 5

Results

This chapter describes: (a) the network approach developed for converting the biclusters into a network of connected and isolated bilcusters (Section 5.1), (b) application of the bicluster network method developed for multi-domain proteins to the RGS family proteins and comparison of the RGS bicluster network with the clusters obtained by other methods including TRIBE-MCL and a phylogenetic method (Section 5.2), (c) application of the bicluster network method to multiple proteomes and its evaluation (Section 5.3), and (d) analysis of overlaping and non-overlaping domain predictions and its effect on proteome biclusters (Section 5.4).

5.1 Protein-Domain Biclusters to Network

Protein-domain biclusters obtained by applying the Bimax algorithm (described in Section 2.5.1) exist as inclusion maximal biclusters, where populations of both proteins and domains are redundant (not unique). However, how these biclusters represent protein relationships in terms of the number of shared domains (similarities) is not clear at this stage. In this section we describe how we generate protein-domain bicluster network that clarifies (a) biclusters that are connected by shared domains, and (b) isolated biclusters that lack any shared domains. The biclusters (bicliques) derived from Bimax are processed to derive these networks as illustrated in Figure 5.1:

1. In this example, the protein set contains 10 proteins (p_1–p_{10}) with 11 domains in various compositions (Figure 5.1A).

2. Bimax generates the inclusion maximal bicliques (Figure 5.1B). Same proteins are included in multiple bicliques as indicated for p_5.

3. Biclusters are refined with respect to their unique domain compositions. From the initial set of biclusters, protein membership across all the bicusters are made unique by removing the overlapping proteins from the bicluster that has the smallest domain set. That is, for every two biclusters in the set, where $B_i = (P_i, D_i)$ and $B_j = (P_j, D_j)$, $\forall i \neq j$ and $P_i \cap P_j \neq \emptyset$, common proteins are removed. Protein p, if $p \in P_i$ and P_j, is retained in the bicluster B_i, if $\mid D_i \mid > \mid D_j \mid$. This is repeated for all pair of biclusters. In Figure 5.1B, for example, protein p_5 marked with boxes is present in three biclusters. After removing two redundant p_5s in Figure 5.1C, only one p_5 is retained in the cluster that has the largest domain set $\{d_1, d_2, d_3\}$.

4. Next, biclusters with common domains are connected to form components of connected biclusters (Figure 5.1D). Two biclusters $B_i = (P_i, D_i)$ and $B_j = (P_j, D_j)$ are connected via a domain d_c, if d_c is present in both D_i and D_j. For example, in Figure 5.1D, three edges marked with yellow arrows to the domain d_{10} connect three biclusters that share the domain. In Figure 5.1D, all

such shared by more than one biclusters are colored grey and every edge that connects two proteins through a shared domain is colored blue.

5. In the final step, each bicluster is converted to a node (Figure 5.1E). Each nodes are connected by an edge where edge weight increases with the number of shared domains between the biclusters.

In this example, biclustering of 10 proteins resulted in seven biclusters (B_1-B_7), two connected clusters (C_1 and C_2) and one isolated (I) bicluster. These connected biclusters could represent multi-domain protein families that are similar to each other with respect to varying numbers of shared domains. These protein subfamilies connected by the shared domains represent complex and larger protein superfamilies.

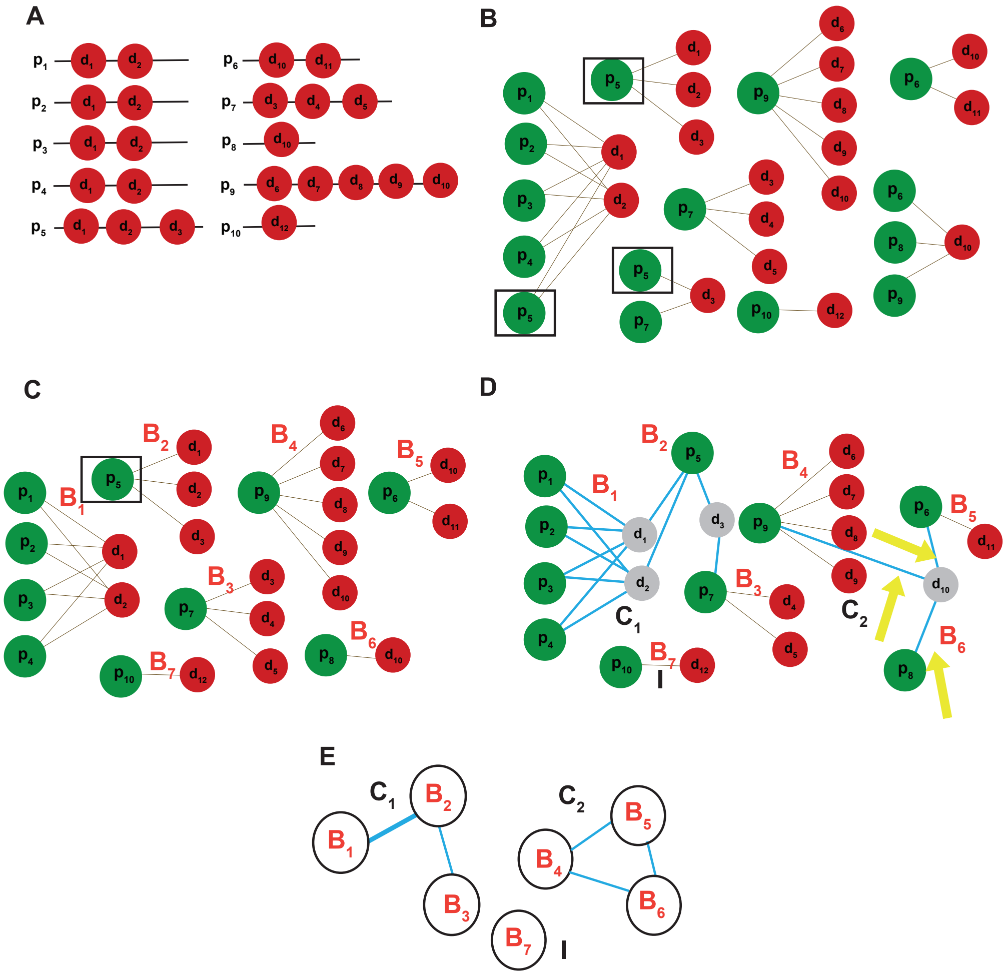


Figure 5.1: Generation of protein-domain biclustering network. (A) Ten proteins with various domain compositions. Domains are represented by red circles. (B) The inclusion maximal bicliques derived from Bimax. Proteins are represented by green circles. Three biclusters, for example, contain a common protein p_5, which is shown in boxes. (C) The seven biclusters (B_1-B_7) after removing the shared redundant proteins. Protein p_5 for example, is retained only in the bicluster B_2 that contains the domain set $\{d_1, d_2, d_3\}$, the largest among the three that included p_5 in the original biclusters. (D) The network containing connected clusters (C_1, C_2) and isolated biclusters (I). For example, the shared domain d_{10} connects the three biclusters $B4, B_5$ and B_6. The edges connecting these biclusters are marked by yellow arrows. All shared domains are colored grey and the edges that connect biclusters are colored blue. (E) Each bicluster is converted into a node. Edges between the biclusters are weighed based on the number of shared domains between them. Wider edges indicate stronger connections between biclusters in terms of shared domains.

5.2 Bicluster Network of RGS Protein family

The biclustering network algorithm was applied to the set of 55 RGS proteins. Figure 5.2 shows how these proteins (green nodes) were grouped into 17 biclusters (B_1–B_{17}).

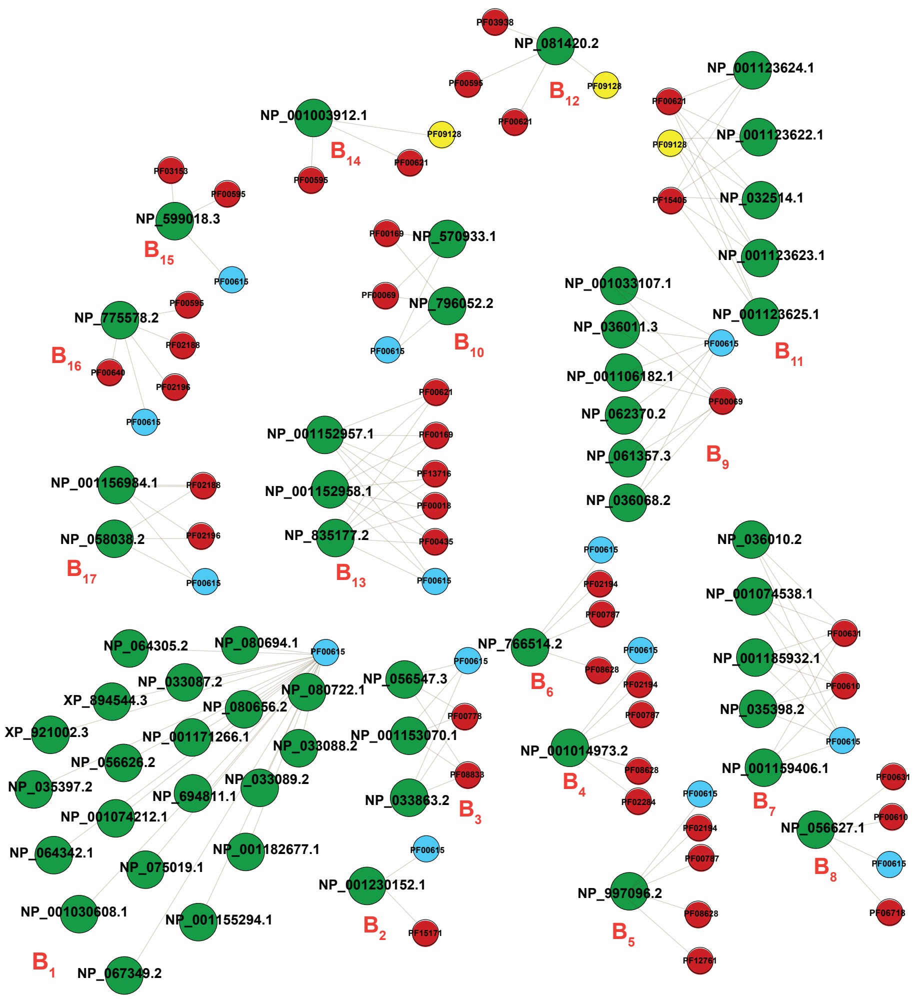


Figure 5.2: Network of Mouse RGS Protein Biclusters. 17 biclusters ($B_1 - B_{17}$) of 55 proteins (green) are biclustered with their respective domains (red). All the proteins contain at least one RGS (blue) or RGS-like (yellow) domain.

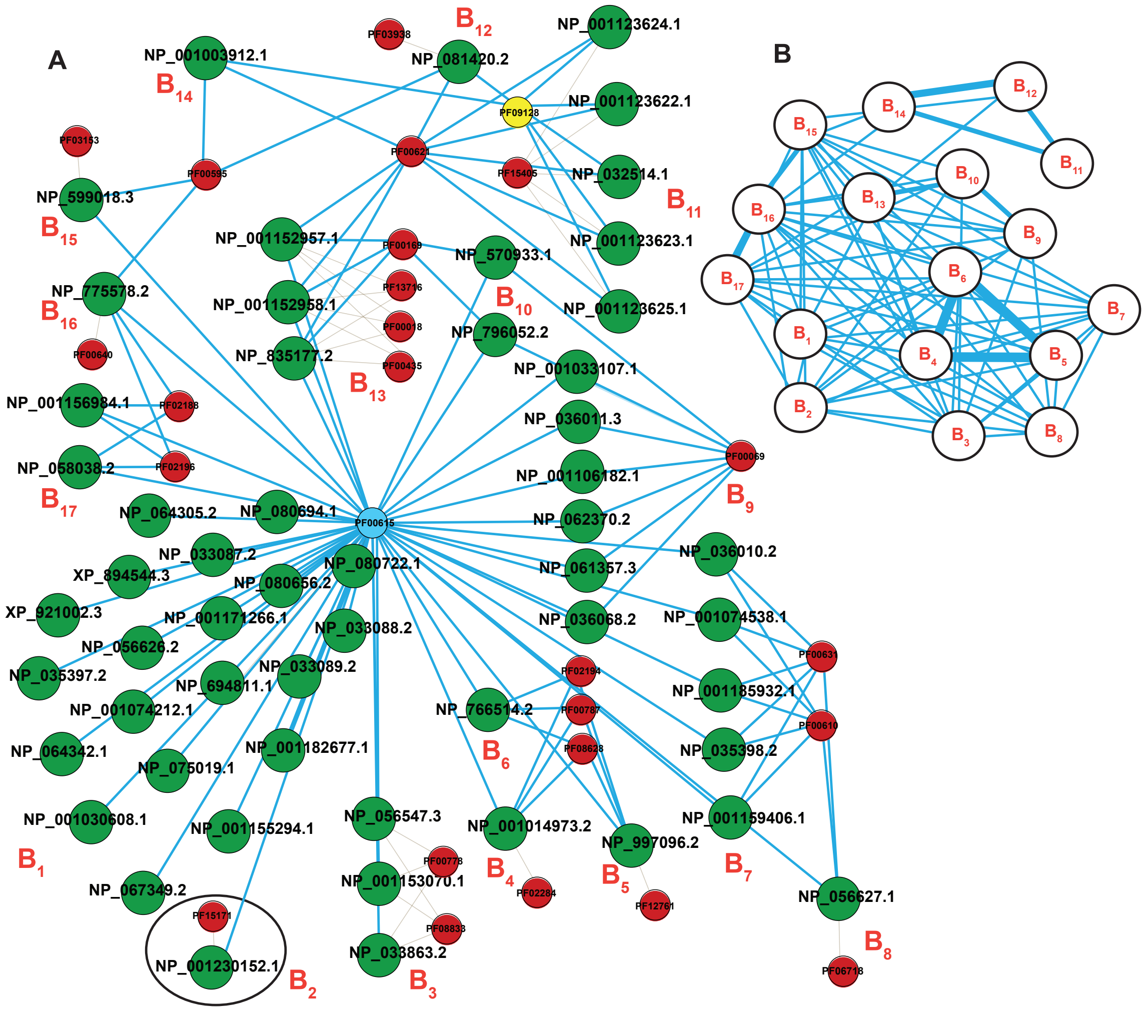


Figure 5.3: Network of Mouse RGS Protein Biclusters. (A) Blue edges connect the biclusters through the common domain(s). (B) Network representing biclusters as nodes. Edges are weighed based on the number of shared domains between the clusters. The biclusters B_4, B_5 and B_6 share four domains and hence are connected by thicker edges compared to the biclusters that share one domain (B_1 and B_2).

Since all RGS proteins have either RGS or RGS-like domain, all proteins in this network are connected to at least the RGS (blue) or RGS-like (yellow) domain. The largest bicluster (B_1, Figure 5.3 A) has 20 proteins where each protein contains the

RGS domain. Even though all the 55 proteins belong to a single RGS protein family, proteins in different clusters are dissimilar to each other with respect to their varying number of domain combinations. For example, 20 proteins containing just one RGS domain form one cluster and are more similar to each other compared to the protein NP_001230152.1 (circled in Figure 5.3)A. This protein has gained a domain PF15171 (Pexin) in addition to the RGS domain and exists in a different cluster. This unique cluster therefore accounts for its dissimilarity from the rest of the 20 proteins. Such a biclustered network of multi-domain proteins therefore provides an explicit account of all the domain gain and/or loss evolutionary events across these proteins.

Bicluster network of proteins and domains are reconstructed in Figure 5.3)B. Each node in this network represents a bicluster and the edge between biclusters are weighed with respect to the number of domains shared between them. For instance, biclusters B_4, B_5 and B_6 share four domains including the RGS domain (PF00615) and therefore the weight of the edges connecting these three biclusters is also four. These relationships between the biclusters provide an insight to the functional coherence of the proteins present in these clusters. Otherwise, as these biclusters share relatively larger number of domains, we speculate that the proteins in these clusters may also perform similar or related functions and hence belong to similar protein subfamily. Gene Ontology (GO) [45] annotation of these proteins were analyzed to support our hypothesis. It is clear that all the three proteins in these biclusters are annotated as "sorting nexin" and share majority of the top GO terms (green, Table5.1). This is a direct evidence that supports the fact these biclustering network besides generating consistent clusters in comparison to other clustering algorithms, it also differentiates the clusters into functionally coherent groups and effectively classifies complex multi-domain proteins into similar protein families.

Table 5.1: GO analysis of three RGS biclusters.

Bicluster	Protein ID	Domains	Annotation	GO terms
B_4	NP_001014973.2	**PF08628**, **PF00787**, **PF02194**, **PF00615**, PF02284	Snx13, sorting nexin 13	GO:0005768, GO:0005769, GO:0006810, GO:0006886, GO:0007154, GO:0008289, GO:0009968, GO:0015031, GO:0016020, GO:0035091, GO:0038032, GO:0043547
B_5	NP_997096.2	**PF08628**, **PF00787**, **PF02194**, **PF00615**, PF12761	Snx25, sorting nexin 25	GO:0003674, GO:0005575, GO:0005768, GO:0006810, GO:0007154, GO:0015031, GO:0016020, GO:0035091, GO:0038032
B_6	NP_766514.2	**PF08628**, **PF00787**, **PF02194**, **PF00615**	Snx14, sorting nexin 14	GO:0003674, GO:0005575, GO:0006810, GO:0007154, GO:0015031, GO:0016020, GO:0016021, GO:0035091, GO:0038032

Domains shared between the biclusters are represented as bold fonts. GO terms with green color are common in all the three biclusters. GO terms with blue color are common in the biclusters B_4 and B_5. GO terms with yellow color are common in the biclusters B_5 and B_6.

5.2.1 Comparison of RGS Protein Biclustering with MCL and Phylogenetic Clustering.

To evaluate the protein-domain biclustering, the same set of RGS proteins were clustered using MCL (Section 4.6.2) and maximum likelihood phylogenetic method (explained in Section 4.6.1). MCL algorithm considers only the most significant similar region between each protein prior to clustering. Similarly, for the maximum likelihood phylogeny method, a multiple sequence alignment is generated based on the domain sequences present in all the proteins. In contrast, biclustering incorporates all the domain information and as a result they provide a clear distinction between proteins with their varying domain composition. For instance, Figure 5.4A shows how seven RGS-like containing proteins are grouped (red circles) into three clusters based on their unique domain compositions. MCL groups the same seven proteins into a single cluster (Figure 5.4B, red circle). These two clustering approaches are largely consistent. Among 55 proteins, 8 of them are grouped 100% consistently (boxed in the figure) in both methods. The clustering based on a phylogenetic method for the same set of seven proteins containing RGS-like domains showed that they are grouped into two clusters (Figure 5.5, arrows) as opposed to three in the biclustering network. A detailed list of the clusters and the respective protein membership is given in the Table 5.2.

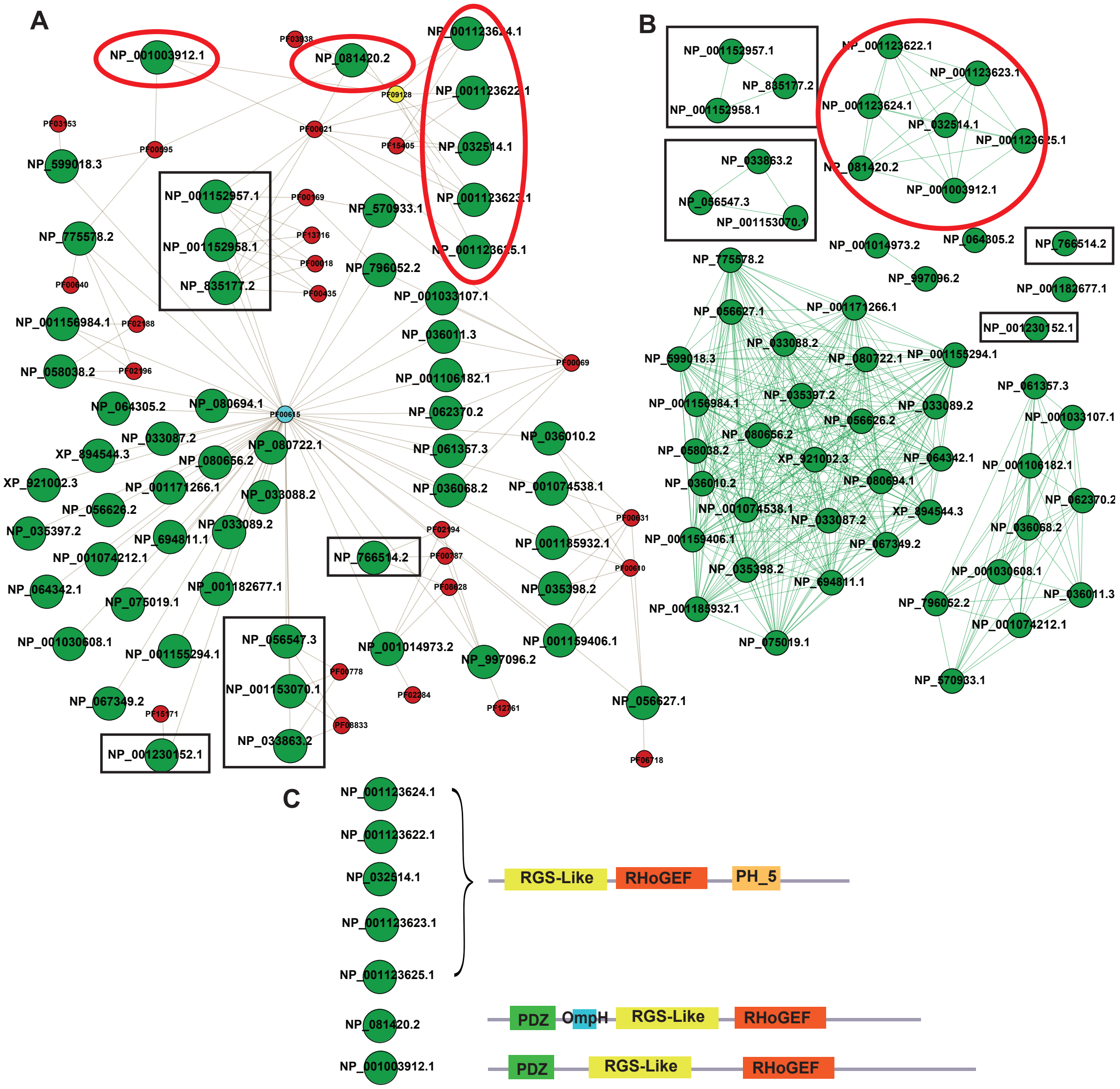


Figure 5.4: Comparison of RGS clusters (A): Clusters of proteins (green) and domains (red) from the biclustering approach. All the proteins contain either the RGS domain (blue) or RGS-like domain (yellow). (B): Clusters from MCL. Proteins (green) in each cluster form a clique. (C): Domain architecture of the seven proteins in the clusters marked by arrows in A and B. Proteins with three unique domain architectures form three clusters in A, and all seven form a single cluster in B. Proteins within the squares are similarly clustered in both A and B. Clusters in red circles shows disparities in both the clustering methods.

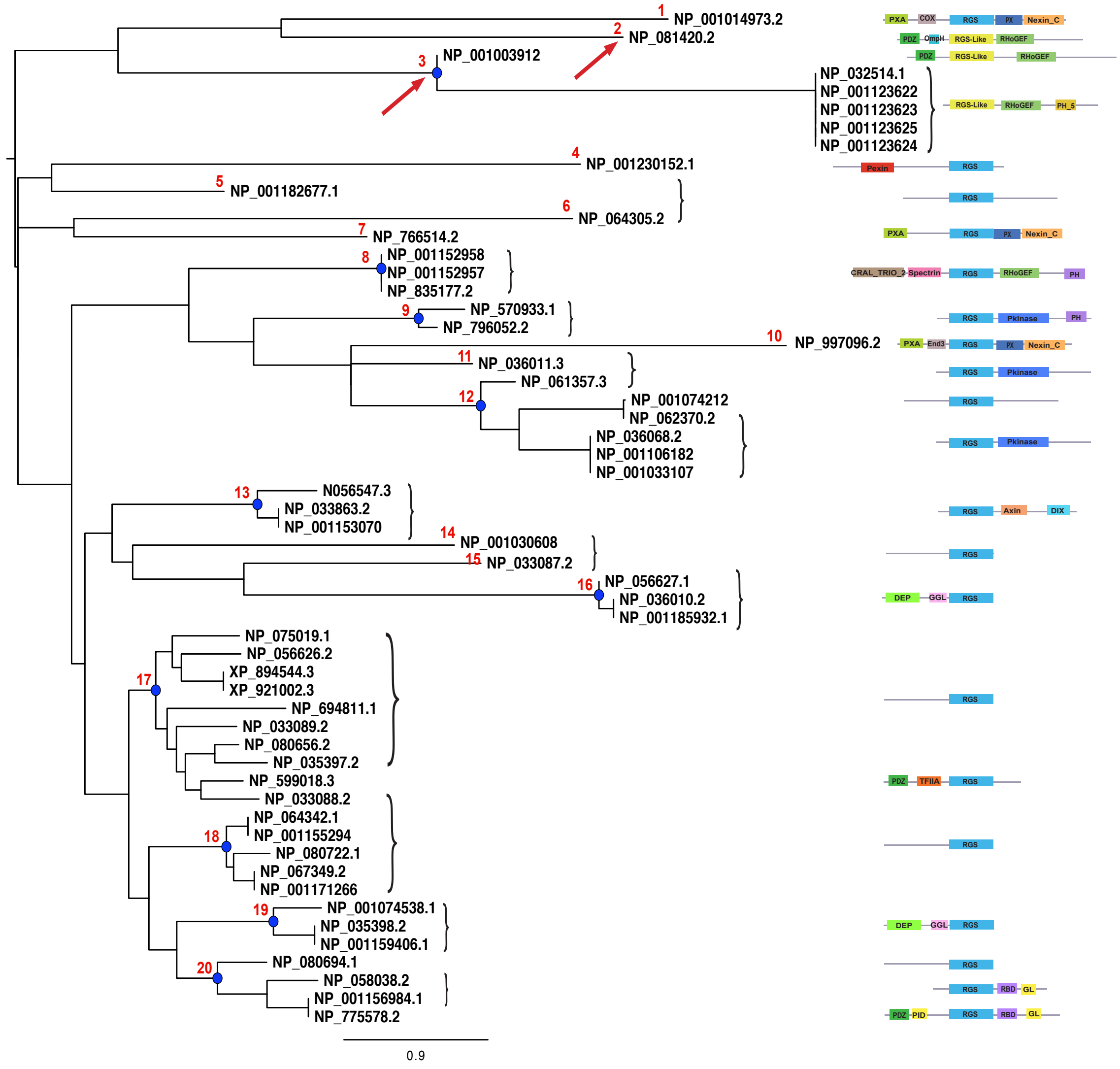


Figure 5.5: Maximum likelihood phylogeny of RGS proteins. Nodes with greater than 70% bootstrap support values are marked with blue circles and they are considered to be clustered. There are 20 clusters derived (marked 1–20). Domain architectures of the proteins are shown to the left. The phylogeny is reconstructed from the multiple sequence alignment of the common domains RGS (cyan) and RGS-like (yellow) domains.

Table 5.2: Results of biclusters, MCL and phylogenetic clusters.

# Bi Cluster	Protein Id	Domain ID	**MCL Clusters**	**Phyml Clusters** [a]
1	NP_033088.2, NP_035397.2, NP_001030608.1, NP_075019.1, NP_067349.2, NP_001182677.1, NP_064305.2, NP_080656.2, NP_001155294.1, NP_056626.2, NP_080694.1, XP_894544.3, NP_001074212.1, NP_033087.2, NP_033089.2, NP_064342.1, NP_080722.1, XP_921002.3, NP_694811.1, NP_001171266.1	PF00615	NP_599018.3, NP_001156984.1, NP_775578.2, NP_058038.2, NP_080694.1, NP_033088.2, NP_033089.2, NP_035397.2, NP_067349.2, NP_080656.2, NP_035398.2, NP_075019.1, NP_056626.2, NP_001159406.1, NP_001171266.1, NP_001074538.1, NP_080722.1, XP_894544.3, XP_921002.3, NP_064342.1, NP_056627.1, NP_001155294.1, NP_033087.2, NP_036010.2, NP_001185932.1, NP_694811.1	NP_075019.1, NP_056626.2, XP_894544.3 XP_921002.3 NP_694811.1, NP_033089.2, NP_080656.2, NP_035397.2, NP_033088.2, NP_599018.3
2	NP_036068.2, NP_061357.3, NP_001106182.1, NP_062370.2, NP_036011.3, NP_001033107.1	PF00069, PF00615	NP_001033107.1, NP_001106182.1, NP_036068.2, NP_570933.1, NP_062370.2, NP_796052.2, NP_001030608.1, NP_061357.3, NP_036011.3, NP_001074212.1	NP_061357.3 NP_001074212.1, NP_062370.2, NP_036068.2, NP_001106182.1, NP_001033107.1
3	NP_001123624.1, NP_001123623.1, NP_032514.1, NP_001123625.1, NP_001123622.1	PF00621, PF09128, PF15405	NP_001003912.1, NP_001123624.1, NP_001123625.1, NP_032514.1, NP_001123623.1, NP_001123622.1, NP_081420.2	NP_001003912.1, NP_032514.1, NP_001123623.1, NP_001123622.1, NP_001123624.1, NP_001123625.1
4	NP_001185932.1, NP_036010.2, NP_001074538.1, NP_001159406.1, NP_035398.2	PF00610, PF00615, PF00631	NP_001152958.1, NP_835177.2, NP_001152957.1	NP_064342.1, NP_001155294.1, NP_080722.1, NP_067349.2, NP_001171266.1
5	NP_001152958.1, NP_835177.2, NP_001152957.1	PF00169, PF00621, PF00435, PF14604, PF07653, PF00615, PF13716, PF00018	NP_033863.2, NP_001153070.1, NP_056547.3	NP_080694.1, NP_058038.2, NP_001156984.1, NP_775578.2
6	NP_056547.3, NP_033863.2, NP_001153070.1	PF00615, PF00778, PF08833	NP_001014973.2, NP_997096.2	NP_001152958.1, NP_835177.2, NP_001152957.1
7	NP_001156984.1, NP_058038.2	PF02188, PF02196, PF00615	NP_001182677.1	NP_056547.3 NP_033863.2, NP_001153070.1
8	NP_796052.2, NP_570933.1	PF00169,PF00615	NP_064305.2	NP_056627.1, NP_036010.2, NP_001185932.1
9	NP_001003912.1	PF00595, PF11333, PF00621, PF09128, PF13180	NP_001230152.1	NP_001074538.1, NP_035398.2, NP_001159406.1
10	NP_081420.2	PF03938, PF00595, PF00621, PF09128, PF13180, PF13476	NP_766514.2	NP_570933.1, NP_796052.2
11	NP_599018.3	PF00595, PF03153, PF00615		NP_001014973.2
12	NP_775578.2	PF02188, PF11470, PF00595, PF00640, PF02196, PF00615, PF13180		NP_081420.2
13	NP_766514.2	PF00615, PF08628, PF00787, PF02194		NP_001230152.1
14	NP_001014973.2	PF08628, PF00787, PF02284, PF02194, PF00615		NP_001182677.1
15	NP_997096.2	PF08628, PF00787, PF02194, PF00615, PF12761		NP_064305.2
16	NP_001230152.1	PF15171, PF00615		NP_766514.2
17	NP_056627.1	PF06718, PF00610, PF00615, PF02234, PF00631		NP_036011.3
				NP_001030608.1
				NP_033087.2
				NP_997096.2

[a]: Proteins within 70% bootstrap support threshold belong to a cluster and rest of them form clusters of singleton proteins.

Table 5.3: Comparison of bicluster, MCL and phylogeny methods based on the Average Maximum Jaccard Index scores.

	Biclusters (17)	MCL Clusters (10)	Phylogeny Clusters (20)[a]
Biclusters	1.00	0.40	0.70
MCL Clusters	0.64	1.00	0.83
Phylogeny Clusters	0.60	0.50	1.00

Total number of clusters from each method is given in the parenthesis.[a]: Proteins within 70% bootstrap support threshold belong to a cluster and rest of them form clusters of singleton proteins.

The number of clusters by biclustering is larger compared to that in MCL. While biclustering network had 17 biclusters, MCL clustered the proteins into 10 clusters. The number of biclusters is proportional to the number of distinct domain compositions present in the data set. However, as MCL clusters the sequences based on the most significant region between them, the number of clusters are not as discrete as in the biclusters. This could be the reason for the number of clusters from the biclusters to be larger. As shown in Table 5.3, the average maximum Jaccard Index (Section 4.7) of MCL clusters against the biclusters is larger ($S_{MCL-BI} = 0.64$) than the relevance of biclusters in MCL ($S_{BI-MCL} = 0.40$). Significant difference in the cluster sizes contribute to the low value of S_{BI-MCL} than S_{MCL-BI}. For the same reason biclusters are more similar to the phylogenetic clusters than to MCL (S_{BI-PHY} = 0.70).

5.3 Biclustering Network on Nine Genomes

5.3.1 Comparison of Biclusters and Markov Clusters

We compared the results of protein clustering by biclustering against MCL algorithm for all the nine proteome data sets. The average maximum Jaccard Index scores were calculated for measuring the similarity of protein clusters in the biclusters to that of respective MCL clusters (S_{BI-MCL}) and vice versa (S_{MCL-BI}) within an E-value threshold of 1. All the biclusters were generated based on the 5% overlap threshold (defined in Section 4.4).

Number of clusters obtained from the biclustering method is larger compared to that from MCL method. For all the nine genomes the number of clusters derived from the biclusters were exceedingly high in comparison to MCL clusters. For example, as given in Table 5.4, total number of biclusters from the mouse genome was 11,763 compared to only 4843 clusters from the MCL method. This increase is observed to be independent of the size of the protein set used. As previously mentioned the number of clusters from the biclustering approach is directly proportional to the number of distinct domain compositions present in the data.

Protein clusters from MCL method are more similar to biclusters than biclusters are to MCL. Irrespective to the genome complexity and domain types (overlap or non-overlap) used, the similarity of protein clusters of MCL to that of the biclusters are much higher (Average $S_{MCL-BI} = 0.70$) compared to the score of biclusters in MCL (Average $S_{BI-MCL} = 0.42$). It is interesting that these features are consistent with what was observed for the RGS biclusters and MCL clusters. The Average Maximum Jaccard Index score is not a symmetric measure. However, we do observe a significant difference in S_{MCL-BI} and S_{BI-MCL}. We speculate that the large difference in the total number of clusters obtained from these two methods could contribute to its difference in the Average maximum Jaccard Index scores.

Table 5.4: Comparison of bicluster and MCL clusters from prokaryotic and eukaryotic genomes.

Proteome datasets		**Bicluster and MCL clusters**			
Species	Domain Type[*]	#Biclusters[a]	#MCL Clusters [b]	S_{BI-MCL}[c]	S_{MCL-BI}[d]
Bacillus subtilis	Overlap	3577	1256	0.33	0.64
	Non-overlap	2753	1256	0.41	0.71
Staphylococcus aureus	Overlap	2268	796	0.34	0.63
	Non-overlap	1863	796	0.40	0.70
Staphylococcus epidermidis	Overlap	2001	690	0.37	0.61
	Non-overlap	1628	690	0.40	0.70
Streptococcus pyogenes	Overlap	1661	652	0.38	0.67
	Non-overlap	1404	652	0.42	0.70
Escherichia coli	Overlap	3598	1402	0.37	0.66
	Non-overlap	2796	1402	0.44	0.72
Yersinia pestis	Overlap	3095	1130	0.35	0.64
	Non-overlap	2446	1130	0.41	0.70
Treponema pallidum	Overlap	812	381	0.44	0.73
	Non-overlap	756	381	0.49	0.76
Drosophila melanogaster	Overlap	11,027	5355	0.42	0.74
	Non-overlap	8772	5355	0.46	0.69
Mus musculus	Overlap	15,188	4843	0.27	0.64
	Non-overlap	11,763	4843	0.33	0.66

[*]: Overlap domain type is the complete set of domain predictions and non-overlap is the domain type that has an overlap length less than 5% of the respective protein length. Detailed definitions are given in the Section 4.4. [a]: Total number of clusters in the biclustering network of the respective genome, [b]: Total number of clusters obtained from the MCL method, [c,d]: Cluster evaluation metric described in Section 4.7.

5.3.2 Network of Protein-Domain Biclusters of Nine Complete Proteomes

The biclustering network algorithm was applied for the complete protein sets of nine genomes (Section 4.5). These nine proteomes varied in their complexity in terms of the number of proteins: from smaller bacterial genomes (average protein number 2917) to much larger eukaryotic genomes (average protein number 29,249).

Figure 5.6 shows a complete network of biclusters of *Staphylococcus aureus*. It clearly shows both isolated (black arrow) and connected biclusters (red arrow) that are present in a network of complete proteome.

Numbers of biclusters increase with genome complexity. In bacteria, the average number of proteins represented in a single bicluster on average is approximately 1. For example, for a total of 2431 proteins of *Staphylococcus aureus*, 1863 biclusters were derived with an average number of proteins in a single cluster of 1.3. In contrast, in eukaryotes (*Mus musculus* and *D. melanogaster*), such a ratio is 3 (Figure 5.7).

Proportion of multi-domain proteins representing complex protein families increases with genome complexity. Interestingly, the proportion of the biclusters that form the components of connected biclusters and isolated clusters also vary from bacteria to eukaryotes. A complete list of the data derived from the biclustering network of each genome is given in Table 5.5. When in *T.pallidum*, the number of connected biclusters are 317, this value increases tremendously to 10,167 connected biclusters in the mouse genome. Proportion of multi-domain proteins forming such connected components also varies with a minimum of 35.31% in *T.pallidum* to a maximum of 82.90% in *M.musculus.* Besides the number of connected proteins these networks also provide information on the number of biclusters (Figure 5.7) obtained with each genome. These numbers also increase with genome complexity.

This proportions of multi-domain proteins within these genomes are consistent with previously established studies. It has been shown that eukaryotes contain a larger proportion (approximately 70%) of multi-domain proteins in comparison to bacteria [9, 10]. The proportions vary as each study has different approaches and domain databases in used for clustering the proteins. In general, multi-domain proteins occur at (a) two-thirds to four-fifths in eukaryotes, and as (b) two-fifths to two-thirds in

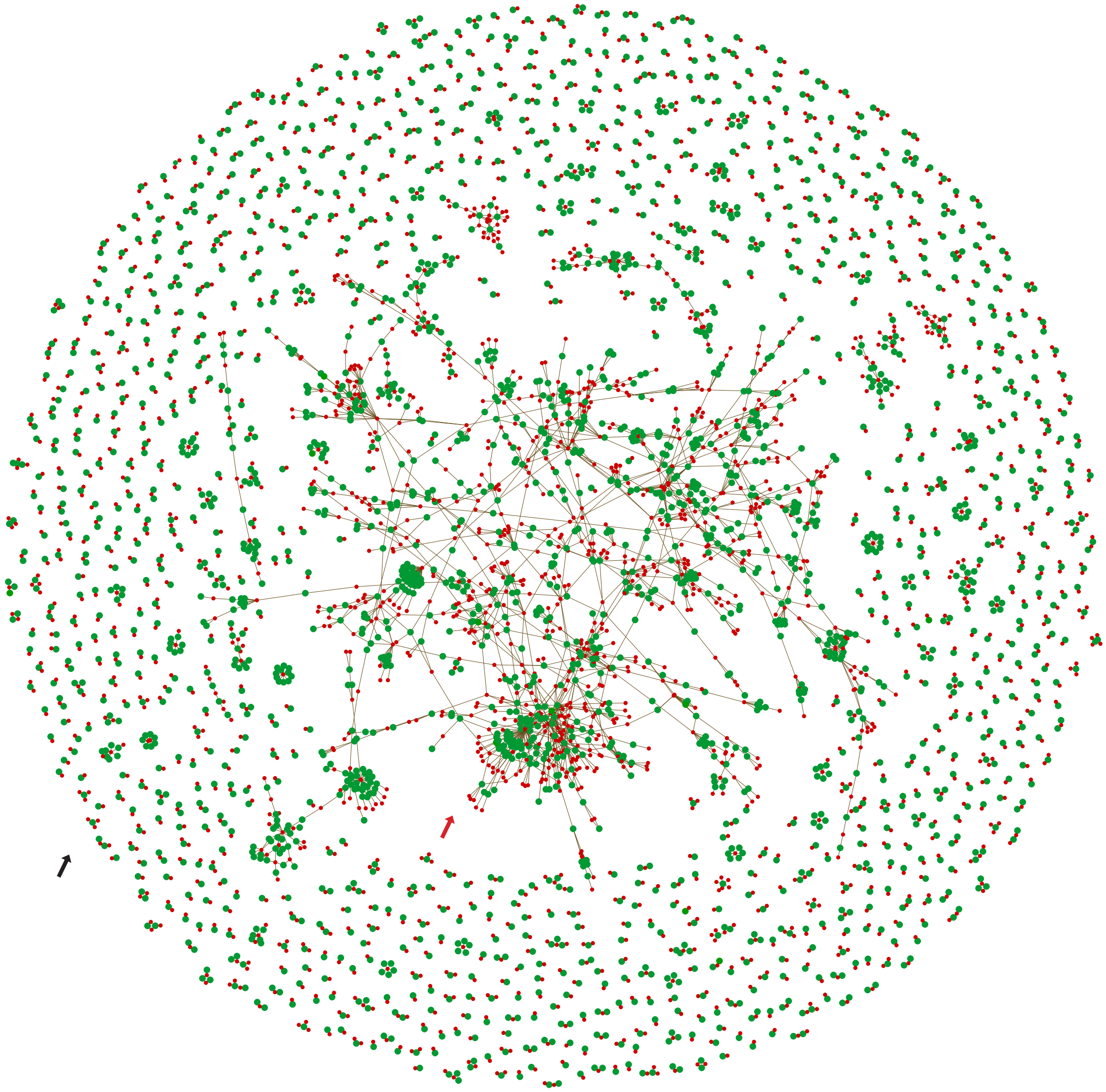


Figure 5.6: Protein-domain network of *Staphylococcus aureus*. Total of 1863 biclusters comprised of 2431 proteins (green) and 2306 domains (red), out of which 948 biclusters form components of connected biclusters (red arrow). The remaining biclusters are isolated (black arrow).

prokaryotes [43]. This is consistent with the data we have obtained and supports the fact that the proportion of multi-domain proteins representing complex protein families are higher in eukayotes compared to that found in bacteria.

Table 5.5: Protein-domain bi-clustering network of prokaryotes and eukaryotes.

Proteome datasets		Biclustered data				
Species	Total proteins	#Proteins	#Domains	#Bi-clusters[a]	#Connected Bi-clusters[b]	#Connected Proteins[c]
Bacillus subtilis	4251	3999	3098	2753	1488	2217 (52.15%)
Staphylococcus aureus	2680	2431	2306	1863	948	1276 (47.61%)
Staphylococcus epidermidis	2268	2129	2086	1628	817	1085 (47.84%)
Streptococcus pyogenes	1964	1764	1784	1404	641	821 (41.80%)
Escherichia coli	4588	4386	3128	2796	1503	2574 (56.10%)
Yersinia pestis	3642	3536	2833	2446	1386	2132 (58.54%)
Treponema pallidum	1028	876	1088	757	317	363 (35.31%)
Drosophila melanogaster	29,217	26,323	6545	8772	7367	22,529 (77.11%)
Mus musculus	29,281	27,979	7654	11,763	10,167	24,275 (82.90%)

[a]: Total number of bi-clusters. [b]: Bi-clusters that are members of connected components in the network. [c]: Proteins that are members of connected components in the network. All the clusters here are based on an E-value threshold of 1 and non-overlap threshold length of 5% of the respective protein (defined in Section 4.4).

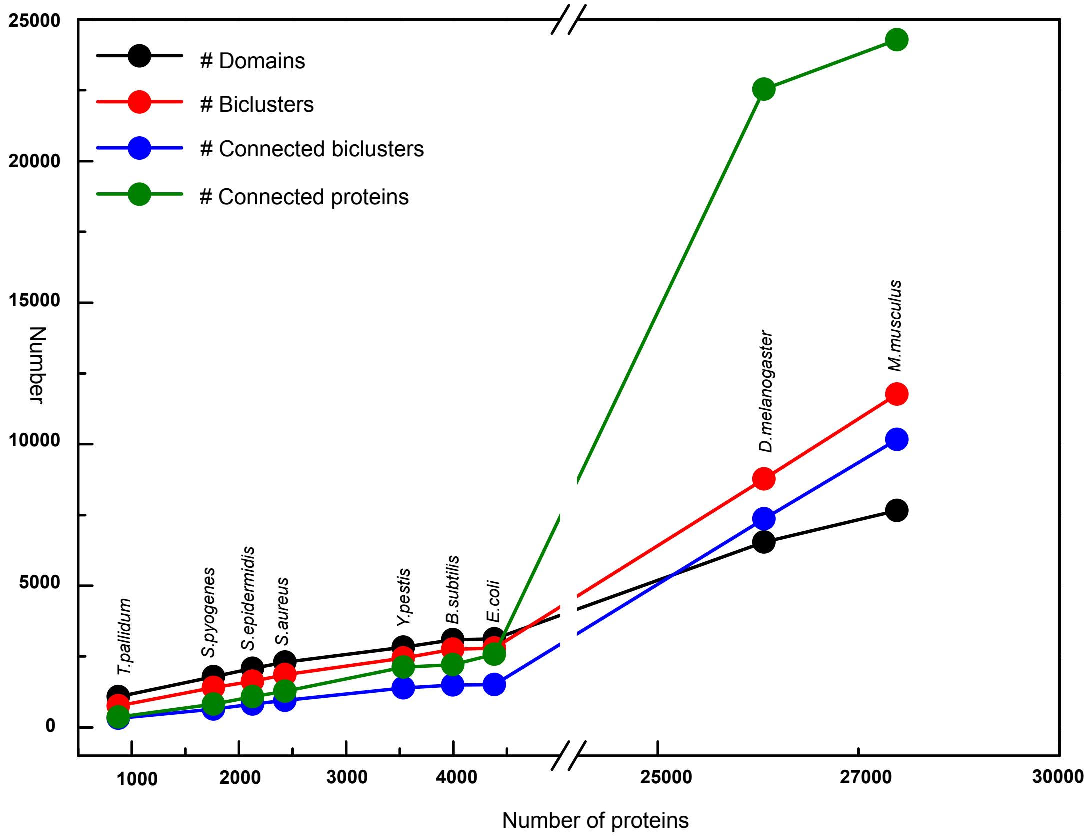


Figure 5.7: Genome complexity (number of proteins) versus domains, biclusters, connected biclusters and connected proteins. X-axis is the total number of proteins with at least one domain prediction. Y-axis is the number of domains, biclusters, connected biclusters or connected proteins. "Connected biclusters" are the number of members of connected components in the network. "Connected proteins" are the number of proteins in the connected components of the network. Each data point represents a species in the increasing order of protein numbers—*Treponema pallidum, Streptococcus pyogenes, Staphylococcus epidermidis, Staphylococcus aureus, Yersinia pestis, Bacillus subtilis, Escherichia coli, Drosophila melanogaster* and *Mus musculus.*

5.4 Overlapping and Non-overlapping Domain Predictions

To analyze the impact of overlapping and non-overlapping domain predictions on biclustering, clustering has been done based on different domain identification strategies described in Section 4.4.

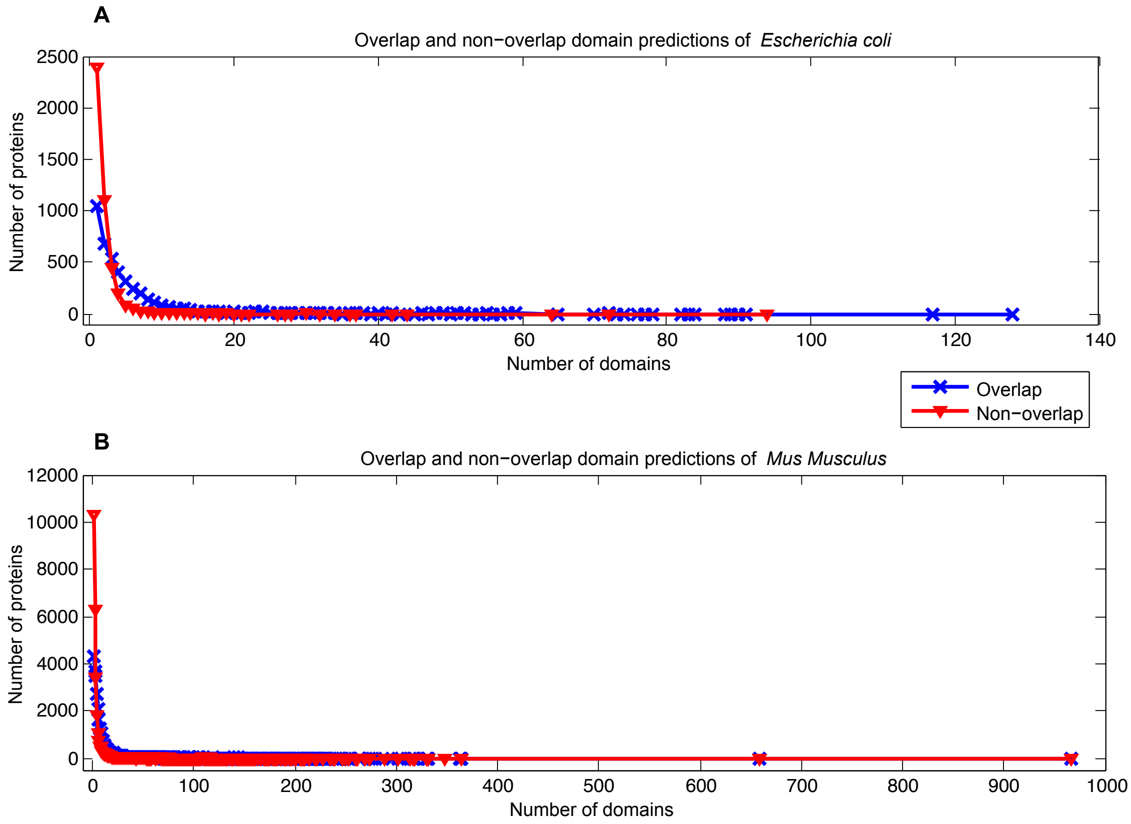


Figure 5.8: Overlapping and non-overlapping domain predictions from the *Escherichia coli* (A) and *Mus musculus* (B) genomes. The number of proteins that have given numbers of domains based on overlapped or non-overlapped prediction is plotted.

Number of biclusters are higher for domain types with overlap predictions.

Number of proteins with single domains are significantly higher when non-overlapping domain predictions are used. As shown in Figure 5.8A, for example, *E. coli* has 2388 proteins containing a single domain for non-overlapping predictions compared to only 1041 such proteins for overlapping domain prediction. For mouse number of proteins with single domain for non-overlapping predictions are 10,290 compared to 4292 for overlap domain prediction. On the other hand, number of domains predicted in a protein are much higher for overlapping domain predictions. One protein in *E. coli*, for example, has 128 domains with overlapping domain prediction and the same protein has only 96 domains when non-overlapping prediction is done. Similar patterns were observed regardless of the genomes.

Table 5.6: Protein-domain bi-clustering details based on overlapping domain predictions.

Proteome datasets		**Biclustered data**					
Species	Total proteins	E-value	#Proteins	#Domains	#Bi-clusters[a]	#Connected Bi-clusters[b]	#Connected Proteins[c]
Bacillus subtilis	4251	0.001	3995	4832	3560	3042	3406 (80.12%)
		1.0	3999	5095	3577	3085	3441 (80.94%)
		10.0	3999	5102	3576	3085	3442 (81.00%)
Staphylococcus aureus	2680	0.001	2430	3675	2258	1854	1989 (74.20%)
		1.0	2431	3898	2268	1888	2017 (75.26%)
		10.0	2431	3902	2268	1888	2017 (75.26%)
Staphylococcus epidermidis	2268	0.001	2129	3480	1994	1636	1744 (76.90%)
		1.0	2129	3696	2001	1665	1766 (77.87%)
		10.0	2129	3697	2001	1665	1766 (77.87%)
Streptococcus pyogenes	1964	0.001	1764	2885	1658	1272	1347(68.85%)
		1.0	1764	2993	1661	1294	1369 (69.70%)
		10.0	1764	2994	1661	1294	1369 (69.70%)
Escherichia coli	4588	0.001	4383	4573	3589	2980	3646 (79.47%)
		1.0	4386	4720	3598	3017	3683 (80.27%)
		10.0	4386	4723	3596	3016	3684 (80.29%)
Yersinia pestis	3642	0.001	3534	4111	3083	2554	2946 (80.90%)
		1.0	3536	4301	3095	2592	2978 (81.80%)
		10.0	3536	4303	3095	2592	2978 (81.80%)
Treponema pallidum	1028	0.001	875	1657	808	495	539 (52.43%)
		1.0	876	1767	812	510	554 (53.89%)
		10.0	876	1769	812	510	554 (53.89%)
Drosophila melanogaster	29,217	0.001	26,284	7931	10,692	9968	24,743 (84.7%)
		1.0	26,323	8277	11,027	10,375	24,957 (85.42%)
		10.0	26,326	8279	11,026	10,375	24,961(85.43%)
Mus musculus	29,281	1.0	27,979	9806	15,188	14,492	26,670 (91.08%)
		10.0	27,983	9806	15,831	15,224	26,928 (92.00%)

[a]: Total number of bi-clusters. [b]: Bi-clusters that are members of connected components in the network. [c]: Proteins that are members of connected components in the network

In comparison to non-overlap domain prediction type, overlap predictions result in larger domain sets. For instance, total number of domains predicted from *Bacillus subtilis* proteome is 3098 for the non-overlapping domain prediction, compared to 5095 for the overlapping domain prediction. Complete list of the domain prediction numbers for both the prediction strategies can be found in Table5.5 (non-overlap) and Table 5.6 (overlap). This in turn has a direct impact on the number of biclusters generated. Number of biclusters derived from the overlapping domain predictions are higher compared to those with non-overlappings. For the RGS protein clusters, overlapping domain predictions have 66 RGS proteins with 24 biclusters (Figure A.1) in contrast to 55 proteins and 17 biclusters when non-overlapping predictions are used (described in Section 5.2). Similar results were observed for the complete proteome data sets used (see Table 5.6). When domain overlaps were allowed, the number of proteins that form components of connected biclusters are significantly higher. For example, as shown in Table 5.5, the proportion of proteins constituting connected components of biclusters was 52.51% for *Bacillus subtilis* with an allowed overlap threshold of 5% length of the protein. The proportion with overlap domain predictions increased up to 80.94% as listed in Table 5.6. Using non-overlapping threshold showed a much higher number of clusters compared to that of 5% overlap threshold (Table A.2).

Varying levels of E-value thresholds also have an interesting effect on the biclusters and on the proportion of proteins found in the connected components of biclusters. The E-value thresholds of 1 and 10 have a moderate effect on the number of proteins representing connected multi-domain protein families. The E-value threshold of 0.001 significantly reduces the number of biclusters and also the proportion of proteins representing complex protein families.

Chapter 6

Discussion

The problem of multi-domain classification into similar protein families has been studied intensively in bioinformatics research. However, this problem still poses numerous challenges especially when it comes to identifying or clustering large scale protein sequences into similar families based on their domain composition. All the proteins that share a single domain do not always imply that they perform the same or related functions [44]. In fact, proteins that contain the same domain architecture or even the same domain composition are functionally more similar than the ones that share single domain. Conventional protein clustering algorithms, such as phylogeny and MCL are based on only domains shared across almost all proteins or the most significant region between the proteins for their clustering process. Therefore, the phylogeny and the clusters derived from MCL fail to represent the exact evolutionary relationships accurately or classify them into specific functionally coherent clusters

In this thesis, a protein classification method for mutli-domain proteins has been developed using protein-domain bicluster network approach. Such networks at a genome level classifies the complete proteome data into groups of similar protein families. It not only addresses the domain combinations of multi-domain proteins, but also accounts for the evolutionary events such as domain gains and losses accurately. In addition to classifying proteins at single protein family (*e.g.*, using the RGS protein sequences), this method can be also applied for classifying proteins at the complete proteome level. Comparison and evaluation of protein biclusters with MCL and phylogenetic clustering methods showed a higher Jaccard Index scores, both at a single protein family and at a complete genome level. One of the direct applications of this method would be its use on large-scale protein databases. Classification of such large protein databases into similar protein families could have numerous analytical and functional applications. One of the caveat in this approach though is its complete dependence on the underlying domain prediction algorithm. Protein clusters generated in this study, for instance, is solely based on the HMMER search algorithm. Therefore, use of more than one or more sensitive domain prediction algorithms such as HHsearch [11] will improve the accuracy of biclusters and eventually on the protein classes derived.

It would also be interesting to define the functional roles of the connected and isolated protein families. This could be accomplished by using the functional annotaion datatabse, *e.g.*, Gene Ontology (GO) [45]. Establishing the functional coherence of the clusters and a functional level analysis would also be a direct application of this method.

Bibliography

[1] T. M. Wilkie and L. Kinch, "New roles for G*alpha* and RGS proteins: communication continues despite pulling sisters apart," *Curr Biol*, vol. 15, no. 20, pp. R843–54, 2005.

[2] A. J. Enright, S. Van Dongen, and C. A. Ouzounis, "An efficient algorithm for large-scale detection of protein families," *Nucleic Acids Res*, vol. 30, no. 7, pp. 1575–84, 2002.

[3] A. Prelic, S. Bleuler, P. Zimmermann, A. Wille, P. Buhlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler, "A systematic comparison and evaluation of biclustering methods for gene expression data," *Bioinformatics*, vol. 22, no. 9, pp. 1122–9, 2006.

[4] G. S. Chang, Y. Hong, K. D. Ko, G. Bhardwaj, E. C. Holmes, R. L. Patterson, and D. B. van Rossum, "Phylogenetic profiles reveal evolutionary relationships within the "twilight zone" of sequence similarity," *Proc Natl Acad Sci U S A*, vol. 105, no. 36, pp. 13474–9, 2008.

[5] UniProtConsortium, "Reorganizing the protein space at the universal protein resource (UniProt)," *Nucleic Acids Res*, vol. 40, no. Database issue, pp. D71–5, 2012.

[6] A. Elofsson and E. L. Sonnhammer, “A comparison of sequence and structure protein domain families as a basis for structural genomics,” *Bioinformatics*, vol. 15, no. 6, pp. 480–500, 1999.

[7] K. Forslund and E. L. Sonnhammer, “Evolution of protein domain architectures,” *Methods Mol Biol*, vol. 856, pp. 187–216, 2012.

[8] J. Lees, C. Yeats, J. Perkins, I. Sillitoe, R. Rentzsch, B. H. Dessailly, and C. Orengo, “Gene3d: a domain-based resource for comparative genomics, functional annotation and protein network analysis,” *Nucleic Acids Res*, vol. 40, no. Database issue, pp. D465–71, 2012.

[9] Z. Wang, R. Cao, and J. Cheng, “Three-level prediction of protein function by combining profile-sequence search, profile-profile search, and domain co-occurrence networks,” *BMC Bioinformatics*, vol. 14 Suppl 3, p. S3, 2013.

[10] G. Apic, J. Gough, and S. A. Teichmann, “Domain combinations in archaeal, eubacterial and eukaryotic proteomes,” *J Mol Biol*, vol. 310, no. 2, pp. 311–25, 2001.

[11] J. Soding, “Protein homology detection by HMM-HMM comparison,” *Bioinformatics*, vol. 21, no. 7, pp. 951–60, 2005.

[12] L. Kannan and W. C. Wheeler, “Maximum parsimony on phylogenetic networks,” *Algorithms Mol Biol*, vol. 7, no. 1, p. 9, 2012.

[13] S. Guindon, J. F. Dufayard, V. Lefort, M. Anisimova, W. Hordijk, and O. Gascuel, “New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of phyml 3.0,” *Syst Biol*, vol. 59, no. 3, pp. 307–21, 2010.

[14] N. M. Luscombe, D. Greenbaum, and M. Gerstein, “What is bioinformatics? a proposed definition and overview of the field,” *Methods Inf Med*, vol. 40, no. 4, pp. 346–58, 2001.

[15] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, “Basic local alignment search tool,” *J Mol Biol*, vol. 215, no. 3, pp. 403–10, 1990.

[16] S. F. Altschul, T. L. Madden, A. A. Schffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, “Gapped BLAST and PSI-BLAST: a new generation of protein database search programs,” *Nucleic Acids Res*, vol. 25, no. 17, pp. 3389–402, 1997.

[17] S. R. Eddy, “Accelerated profile HMM searches,” *PLoS Comput Biol*, vol. 7, no. 10, p. e1002195, 2011.

[18] S. R. Eddy, “Profile hidden markov models,” *Bioinformatics*, vol. 14, no. 9, pp. 755–63, 1998.

[19] M. Punta, P. C. Coggill, R. Y. Eberhardt, J. Mistry, J. Tate, C. Boursnell, N. Pang, K. Forslund, G. Ceric, J. Clements, A. Heger, L. Holm, E. L. Sonnhammer, S. R. Eddy, A. Bateman, and R. D. Finn, “The pfam protein families database,” *Nucleic Acids Res*, vol. 40, no. Database issue, pp. D290–301, 2012.

[20] S. Karlin and S. F. Altschul, “Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes,” *Proc Natl Acad Sci U S A*, vol. 87, no. 6, pp. 2264–8, 1990.

[21] S. R. Eddy, “A probabilistic model of local sequence alignment that simplifies statistical significance estimation,” *PLoS Comput Biol*, vol. 4, no. 5, p. e1000069, 2008.

[22] V. S. Dongen, “Graph clustering by flow simulation,” *PhD thesis, University of Utrecht*, 2000.

[23] J. A. Hartigan, “Direct clustering of a data matrix,” *JASA*, vol. 67, pp. 123–129, 1972.

[24] Y. Cheng and G. M. Church, “Biclustering of expression data,” *Proc Int Conf Intell Syst Mol Biol*, vol. 8, pp. 93–103, 2000.

[25] S. C. Madeira and A. L. Oliveira, “Biclustering algorithms for biological data analysis: a survey,” *IEEE/ACM Trans Comput Biol Bioinform*, vol. 1, no. 1, pp. 24–45, 2004.

[26] A. D. Pablo, O. Fabrcio, M. F. Hamilton, and F. J. V. Zuben., “Applying biclustering to text mining: An immune-inspired approach,” *Lecture Notes in Computer Science*, vol. 4628, pp. 83–94, 2007.

[27] R. Peeters, “The maximum edge biclique problem is NP-complete,” *Discrete Applied Mathematics*, vol. 131, no. 3, pp. 651–654, 2003.

[28] K. M. Murali and S. Kasif, “Extracting conserved gene expression motifs from gene expression data,” *Pac Symp Biocomputing*, no. 8, pp. 77–88, 2003.

[29] A. Tanay, R. Sharan, and R. Shamir, “Discovering statistically significant biclusters in gene expression data,” *Bioinformatics*, vol. 18 Suppl 1, pp. S136–44, 2002.

[30] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini, “Discovering local structure in gene expression data: the order-preserving sub-matrix problem,” *In Proceedings of the 6th Annual Conference on Computtional Biology*, pp. 49–57, 2002.

[31] I. Cohen-Gihon, R. Nussinov, and R. Sharan, “Comprehensive analysis of co-occurring domain sets in yeast proteins,” *BMC Genomics*, vol. 8, p. 161, 2007.

[32] J. Nacher, T. Ochiai, M. Hayashida, and T. Akutsu, “A bipartite graph based model of protein domain networks,” *Social Inforamtics and Telecommunications Enginnering*, vol. 4, pp. 525–535, 2009.

[33] Y. Ye and A. Godzik, “Comparative analysis of protein domain organization,” *Genome Res*, vol. 14, no. 3, pp. 343–53, 2004.

[34] S. Wuchty and E. Almaas, “Evolutionary cores of domain co-occurrence networks,” *BMC Evol Biol*, vol. 5, p. 24, 2005.

[35] S. K. Kummerfeld and S. A. Teichmann, “Protein domain organisation: adding order,” *BMC Bioinformatics*, vol. 10, p. 39, 2009.

[36] R. Rentzsch and C. A. Orengo, “Protein function prediction using domain families,” *BMC Bioinformatics*, vol. 14 Suppl 3, p. S5, 2013.

[37] S. Liang, D. Zheng, D. M. Standley, H. Guo, and C. Zhang, “A novel function prediction approach using protein overlap networks,” *BMC Syst Biol*, vol. 7, p. 61, 2013.

[38] H. Fang and J. Gough, “A domain-centric solution to functional genomics via dcgo predictor,” *BMC Bioinformatics*, vol. 14 Suppl 3, p. S9, 2013.

[39] A. Marchler-Bauer, J. B. Anderson, C. DeWeese-Scott, N. D. Fedorova, L. Y. Geer, S. He, D. I. Hurwitz, J. D. Jackson, A. R. Jacobs, C. J. Lanczycki, C. A. Liebert, C. Liu, T. Madej, G. H. Marchler, R. Mazumder, A. N. Nikolskaya, A. R. Panchenko, B. S. Rao, B. A. Shoemaker, V. Simonyan, J. S. Song, P. A. Thiessen, S. Vasudevan, Y. Wang, R. A. Yamashita, J. J. Yin, and S. H. Bryant,

"Cdd: a curated entrez database of conserved domain alignments," *Nucleic Acids Res*, vol. 31, no. 1, pp. 383–7, 2003.

[40] B. Morgenstern, "Dialign: multiple DNA and protein sequence alignment at bibiserv," *Nucleic Acids Res*, vol. 32, no. Web Server issue, pp. W33–6, 2004.

[41] S. Wuchty, "Scale-free behavior in protein domain networks," *Mol Biol Evol*, vol. 18, no. 9, pp. 1694–702, 2001.

[42] K. Katoh and D. M. Standley, "Mafft multiple sequence alignment software version 7: improvements in performance and usability," *Mol Biol Evol*, vol. 30, no. 4, pp. 772–80, 2013.

[43] J. H. Han, S. Batey, A. A. Nickson, S. A. Teichmann, and J. Clarke, "The folding and evolution of multidomain proteins," *Nat Rev Mol Cell Biol*, vol. 8, no. 4, pp. 319–30, 2007.

[44] S. Henikoff, E. A. Greene, S. Pietrokovski, P. Bork, T. K. Attwood, and L. Hood, "Gene families: the taxonomy of protein paralogs and chimeras," *Science*, vol. 278, no. 5338, pp. 609–14, 1997.

[45] S. Carbon, A. Ireland, C. J. Mungall, S. Shu, B. Marshall, and S. Lewis, "Amigo: online access to ontology and annotation data," *Bioinformatics*, vol. 25, no. 2, pp. 288–9, 2009.

Appendix A

Supplementary Materials

A.1 Tables

Table A.1: RGS protein IDs with their domain compositions.

Protein ID	Domain[a]
NP_033088.2, NP_035397.2, NP_001030608.1, NP_075019.1, NP_067349.2, NP_001182677.1, NP_064305.2, NP_080656.2, NP_001155294.1, NP_056626.2, NP_080694.1, XP_894544.3, NP_001074212.1, NP_033087.2, NP_033089.2, NP_064342.1, NP_080722.1, XP_921002.3, NP_694811.1, NP_001171266.1	PF00615
NP_036068.2, NP_061357.3, NP_001106182.1, NP_062370.2, NP_036011.3, NP_001033107.1	PF00069, PF00615
NP_001123624.1, NP_001123623.1, NP_032514.1, NP_001123625.1, NP_001123622.1	PF00621, PF09128, PF15405
NP_001185932.1, NP_036010.2, NP_001074538.1, NP_001159406.1, NP_035398.2	PF00610, PF00615, PF00631
NP_001152958.1, NP_835177.2, NP_001152957.1	PF00169, PF00621, PF00435, PF14604, PF07653, PF00615, PF13716, PF00018
NP_056547.3, NP_033863.2, NP_001153070.1	PF00615, PF00778, PF08833
NP_001156984.1, NP_058038.2	PF02188, PF02196, PF00615
NP_796052.2, NP_570933.1	PF00169,PF00615
NP_001003912.1	PF00595, PF11333, PF00621, PF09128, PF13180
NP_081420.2	PF03938, PF00595, PF00621, PF09128, PF13180, PF13476
NP_599018.3	PF00595, PF03153, PF00615
NP_775578.2	PF02188, PF11470, PF00595, PF00640, PF02196, PF00615, PF13180
NP_766514.2	PF00615, PF08628, PF00787, PF02194
NP_001014973.2	PF08628, PF00787, PF02284, PF02194, PF00615
NP_997096.2	PF08628, PF00787, PF02194, PF00615, PF12761
NP_001230152.1	PF15171, PF00615
NP_056627.1	PF06718, PF00610, PF00615, PF02234, PF00631
*NP_598838.3	PF14605, PF00642, PF13893, PF10337, PF10978, PF00076, PF14259, PF01480, PF08777
*NP_001207426.1, XP_987134.2, XP_003945709.1, NP_061217.3, NP_001160118.1, XP_003945710.1, XP_001474919.1, NP_001207427.1, NP_001160073.1, XP_986693.2	PF04803

[a]: Domain ID derived from the HMMER prediction for each sequence against the Pfam Database. *: Sequences that are present in overlapping domain prediction set only.

Table A.2: Protein-domain bi-clustering details based on non-overlapping* domain predictions.

Proteome datasets		Biclustered data					
Species	Total proteins	E-value	#Proteins	#Domains	#Bi-clusters[a]	#Connected Bi-clusters[b]	#Connected Proteins[c]
Bacillus subtilis	4251	0.001	3995	2700	2498	1050	1816 (42.72%)
		1.0	3999	2822	2616	1237	1975 (46.46%)
		10.0	3999	2886	2674	1324	2049 (48.20%)
Staphylococcus aureus	2680	0.001	2430	1997	1704	636	1007 (37.57%)
		1.0	2431	2097	1772	769	1128 (42.09%)
		10.0	2431	2133	1810	831	1162 (43.36%)
Staphylococcus epidermidis	2268	0.001	2129	1803	1518	584	868 (38.27%)
		1.0	2129	1905	1580	715	992 (43.74%)
		10.0	2129	1951	1612	753	1007 (44.40%)
Streptococcus pyogenes	1964	0.001	1764	1564	1322	485	700 (35.64%)
		1.0	1764	1645	1361	557	754 (38.39%)
		10.0	1764	1683	1390	605	783 (39.89%)
Escherichia coli	4588	0.001	4383	2819	2550	1069	2125 (46.32%)
		1.0	4386	2909	2644	1229	2291 (49.93%)
		10.0	4386	2951	2699	1313	2360 (51.44%)
Yersinia pestis	3642	0.001	3534	2540	2238	1013	1816 (49.86%)
		1.0	3536	2636	2330	1158	1929 (52.97%)
		10.0	3536	2680	2380	1224	1963 (53.90%)
Treponema pallidum	1028	0.001	875	934	730	246	308 (29.96%)
		1.0	876	983	742	267	322 (31.32%)
		10.0	876	997	748	281	331 (32.20%)
Drosophila melanogaster	29,217	0.001	26,284	5571	6957	5034	20,261 (69.35%)
		1.0	26,323	5931	7804	6158	21,661 (74.14%)
		10.0	26,326	6029	8138	6554	21,925 (75.04%)
Mus musculus	29,281	0.001	27,962	6477	8927	6737	22,127 (75.57%)
		1.0	27,979	6885	10,026	8127	23,215 (79.28%)
		10.0	27,983	6885	10,028	8129	23,217 (79.29%)

$*$: Non-overlapping threshold, where no overlap is allowed. [a]: Total number of bi-clusters. [b]: Bi-clusters that are members of connected components in the network. [c]: Proteins that are members of connected components in the network

A.2 Figures

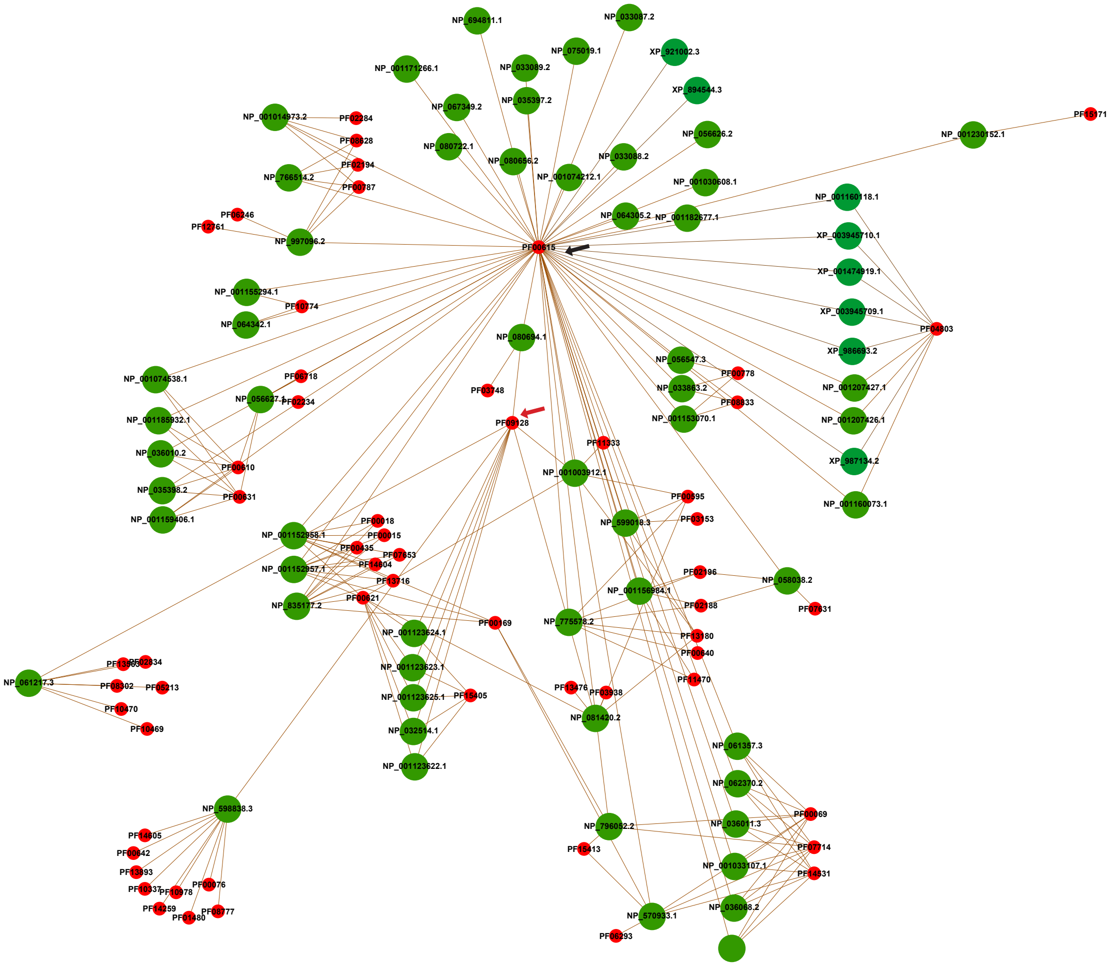
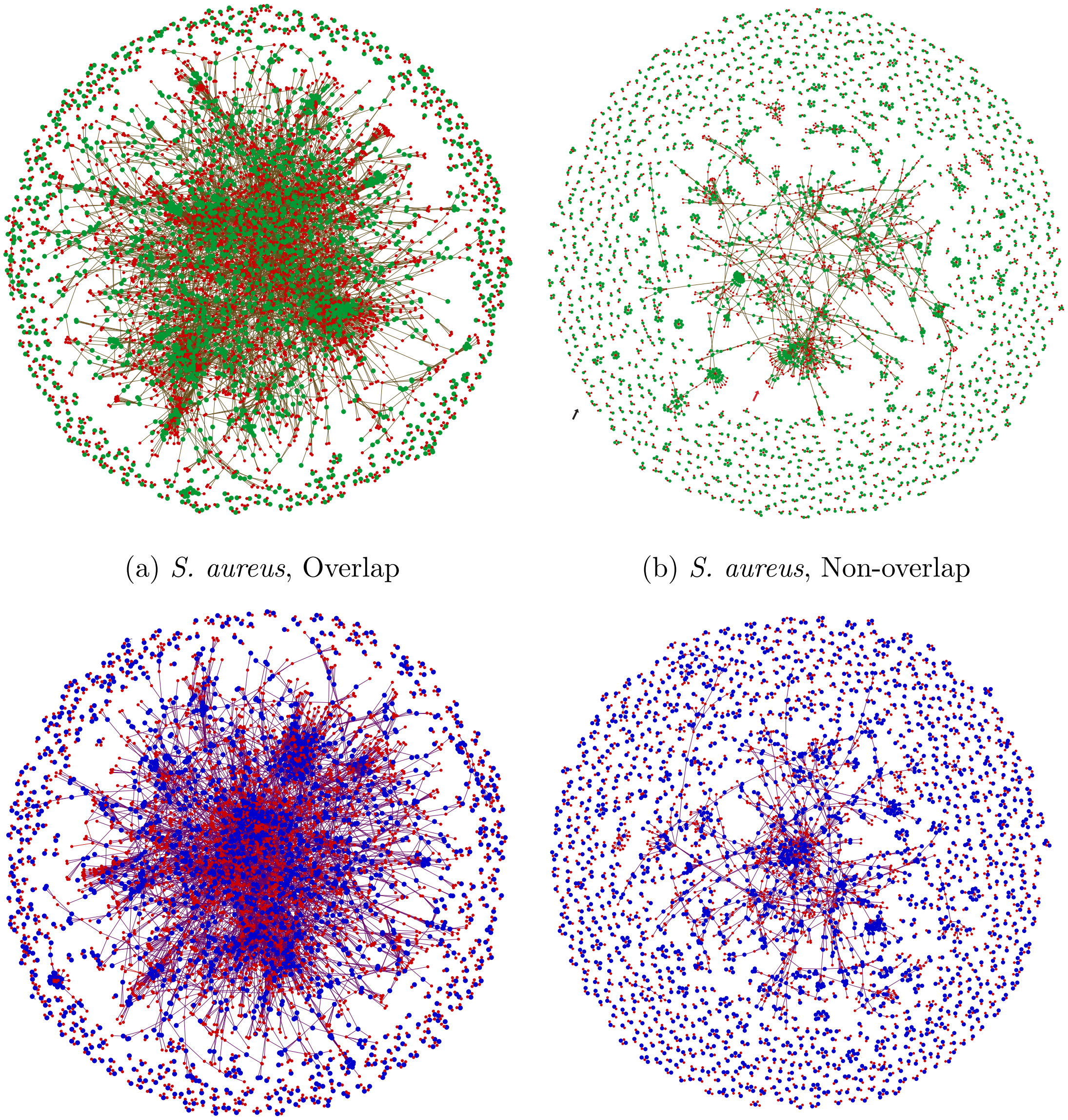


Figure A.1: Protein-domain biclustering network of RGS proteins based on overlapping domain prediction type. Proteins are colored green and domain nodes are colored red. Black and red arrow shows the RGS and RGS-like domains, respectively.



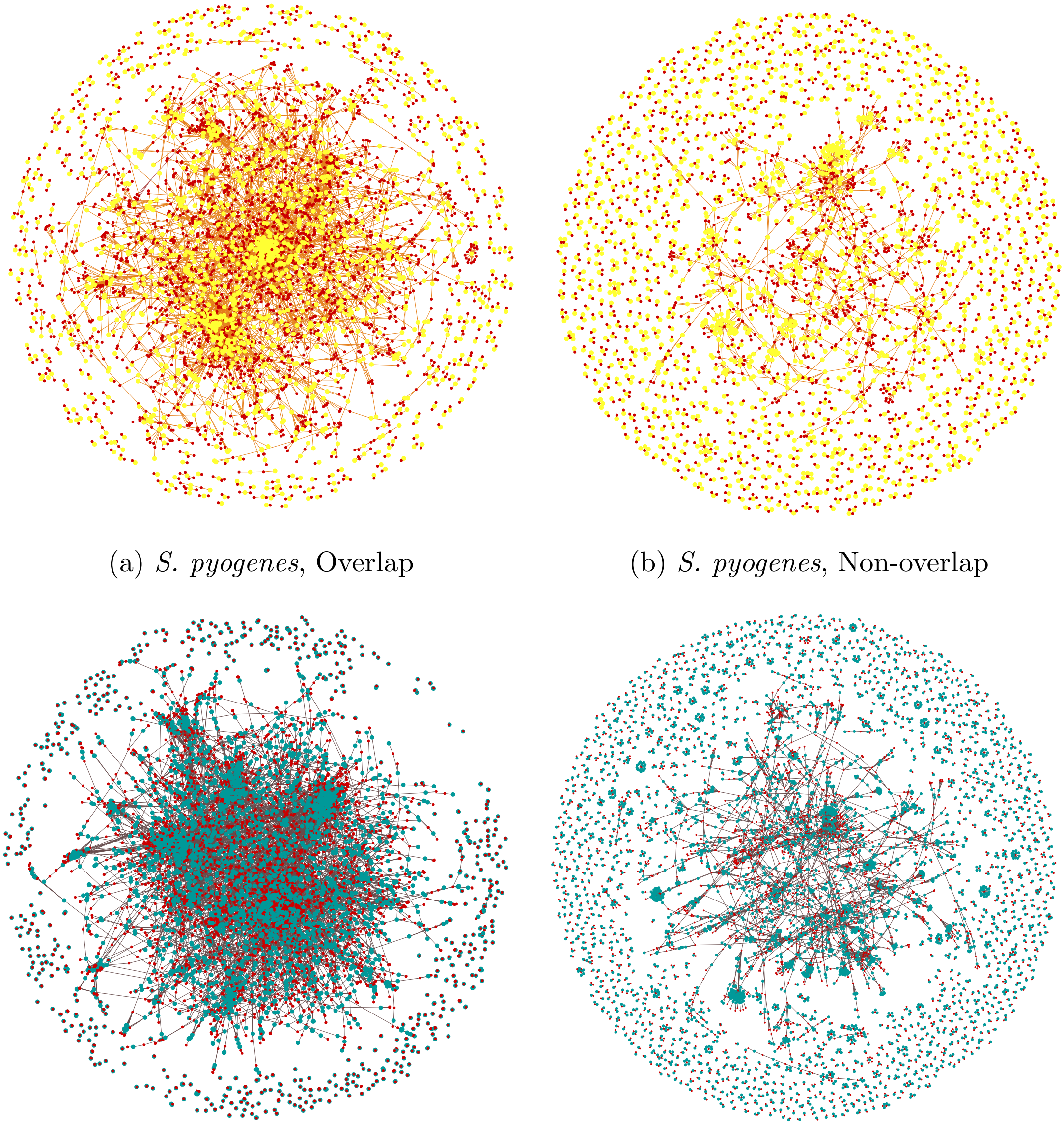
(a) *S. aureus*, Overlap

(b) *S. aureus*, Non-overlap

(c) *S. epidermidis*, Overlap

(c) *S. epidermidis*, Non-overlap

Figure A.2: Comparison of biclustered genome network profile. (a) and (b) Biclustered protein-domain (green-red) network profiles of the genome *Staphylococcus aureus* with overlap and non-overlap domain predictions, respectively. (c) and (d) Biclustered protein-domain (blue-red) network profiles of the genome *Staphylococcus epidermidis* with overlap and domain predictions. Non-overlap threshold — 5% of the protein length.



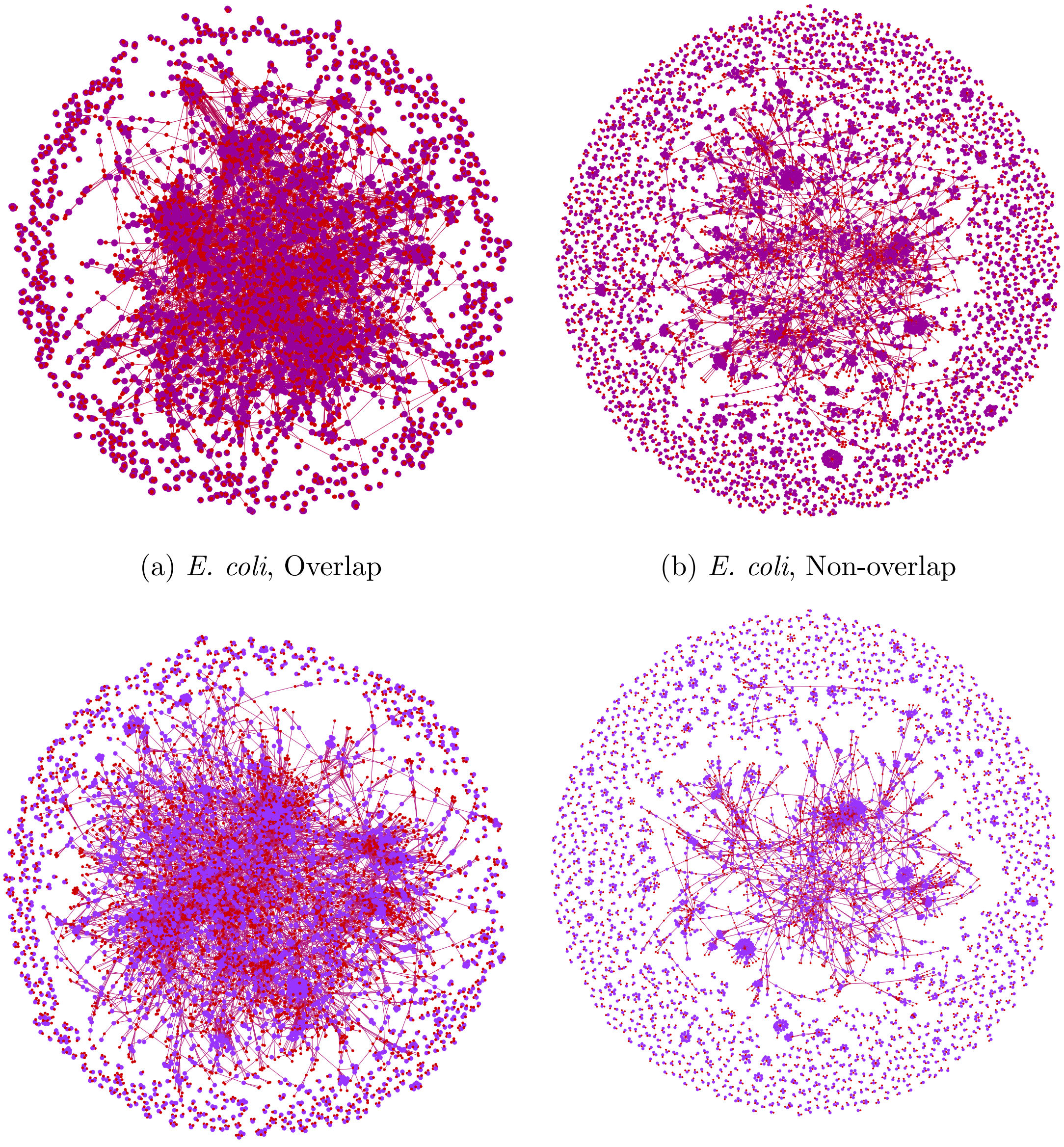
(a) *S. pyogenes*, Overlap

(b) *S. pyogenes*, Non-overlap

(c) *B. subtilis*, Overlap

(c) *B. subtilis*, Non-overlap

Figure A.3: Comparison of biclustered genome network profile. (a) and (b) Biclustered protein-domain (yellow-red) network profiles of the genome *Streptococcus pyogenes* with overlap and non-overlap domain predictions, respectively. (c) and (d) Biclustered protein-domain (cyan-red) network profiles of the genome *Bacillus subtilis* with overlap and domain predictions. Non-overlap threshold — 5% of the protein length.



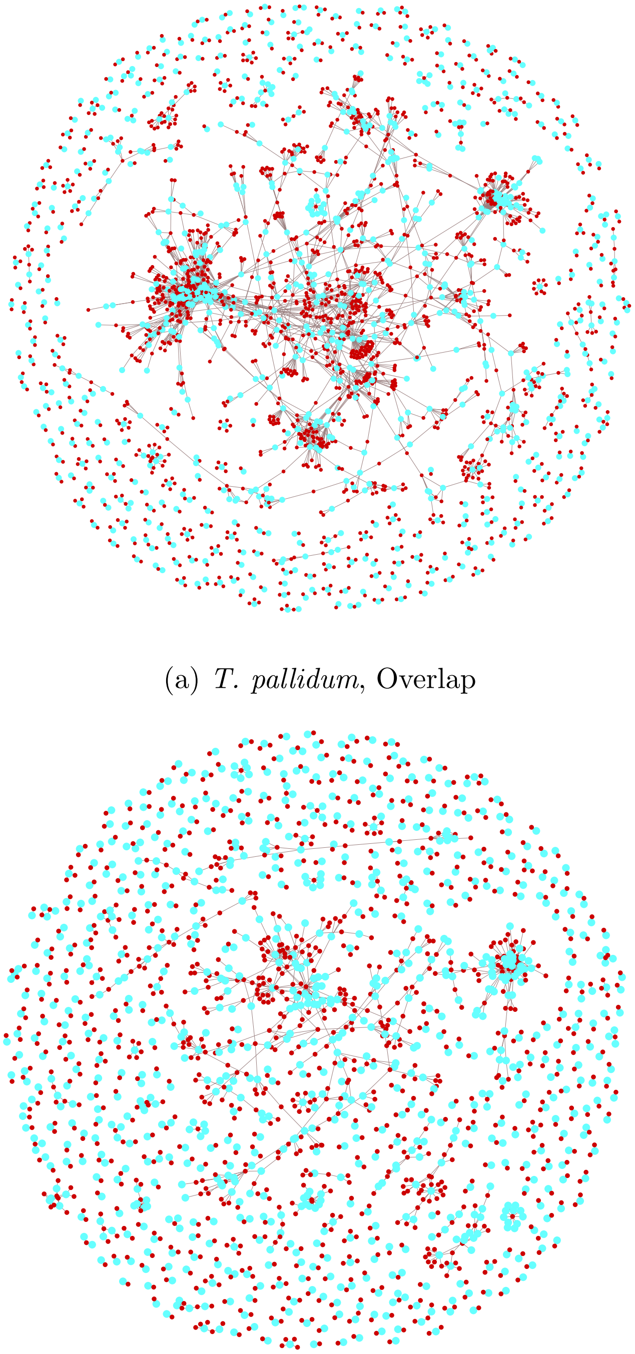
(a) *E. coli*, Overlap

(b) *E. coli*, Non-overlap

(c) *Y. pestis*, Overlap

(c) *Y. pestis*, Non-overlap

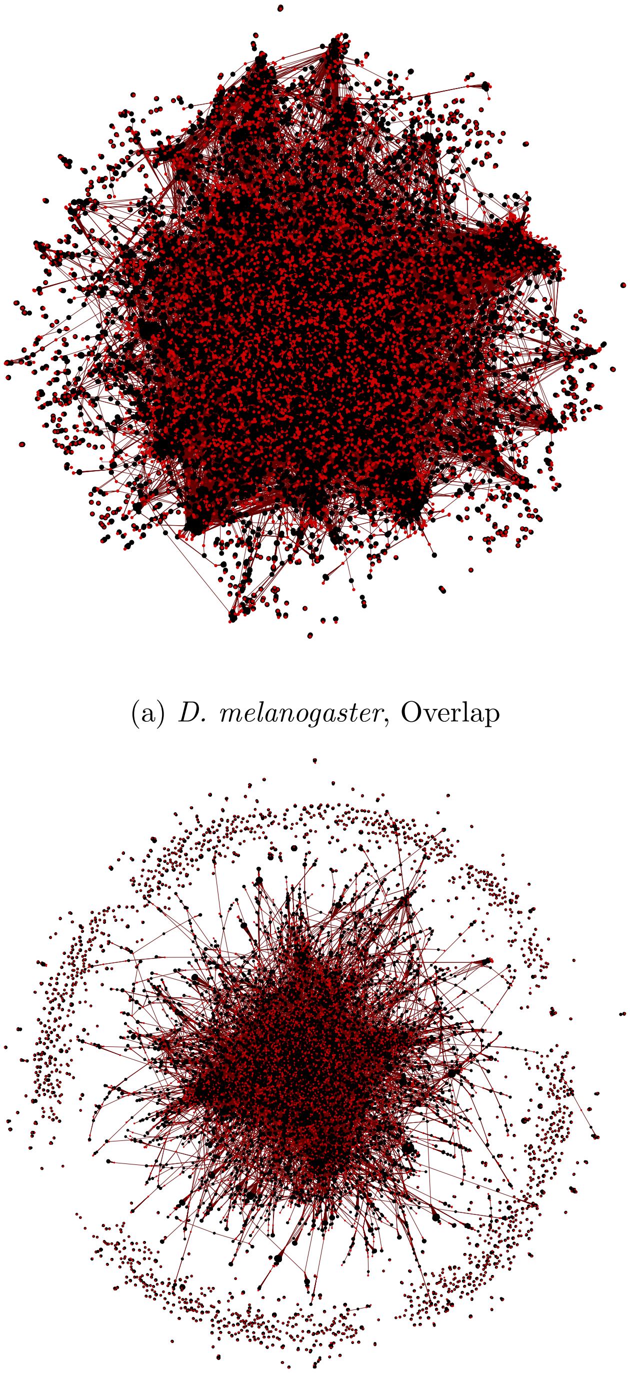
Figure A.4: Comparison of biclustered genome network profile. (a) and (b) Biclustered protein-domain (pink-red) network profiles of the genome *Escherichia coli* with overlap and non-overlap domain predictions, respectively. (c) and (d) Biclustered protein-domain (purple-red) network profiles of the genome *Yersinia pestis* with overlap and domain predictions. Non-overlap threshold — 5% of the protein length.



(a) *T. pallidum*, Overlap

(b) *T. pallidum*, Non-overlap

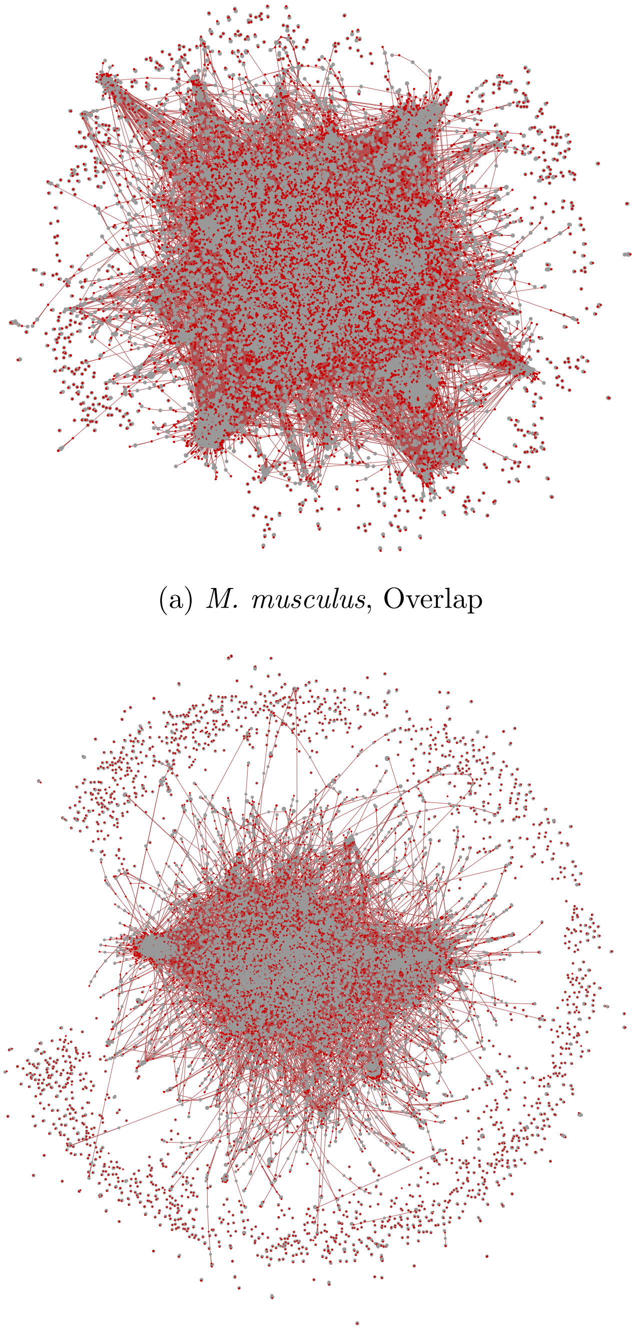
Figure A.5: Comparison of biclustered genome network profile. (a) and (b) Biclustered protein-domain (blue-red) network profiles of the genome *Treponema pallidum* with overlap and non-overlap domain predictions, respectively.



(a) *D. melanogaster*, Overlap

(b) *D. melanogaster*, Non-overlap

Figure A.6: Comparison of biclustered genome network profile. (a) and (b) Biclustered protein-domain (black-red) network profiles of the genome *Drosophila melanogaster* with overlap and non-overlap domain predictions, respectively. Non-overlap threshold — 5% of the protein length.



(a) *M. musculus*, Overlap

(b) *M. musculus*, Non-overlap

Figure A.7: Comparison of biclustered genome network profile. (a) and (b) Biclustered protein-domain (grey-red) network profiles of the genome *Mus musculus* with overlap and non-overlap domain predictions. Non-overlap threshold — 5% of the protein length.