Chapter 18

A SURVEY OF GRAPH MINING TECHNIQUES FOR BIOLOGICAL DATASETS

S. Parthasarathy
The Ohio State University
2015 Neil Ave, DL395, Columbus, OH
srini@cse.ohio-state.edu

S. Tatikonda
The Ohio State University
2015 Neil Ave, DL395, Columbus, OH
tatikond@cse.ohio-state.edu

D. Ucar
The Ohio State University
2015 Neil Ave, DL395, Columbus, OH
ucar@cse.ohio-state.edu

Abstract

Mining structured information has been the source of much research in the data mining community over the last decade. The field of bioinformatics has emerged as important application area in this context. Examples abound ranging from the analysis of protein interaction networks to the analysis of phylogenetic data. In this article we survey the principal results in the field examining them both from the algorithmic contributions and applicability in the domain in question. We conclude this article with a discussion of the key results and identify some interesting directions for future research.

Keywords: Graph Mining, Tree Mining, Biological Networks, Community Discovery

C.C. Aggarwal and H. Wang (eds.), *Managing and Mining Graph Data*,
Advances in Database Systems 40, DOI 10.1007/978-1-4419-6045-0_18,

1. Introduction

Advances in data collection and storage technology have led to a proliferation of structured information available to organizations and individuals. This information is often also available to the user in a myriad of formats and across multiple media. This is especially true in the vibrant field of bioinformatics where an increasing large number of problems are represented in structured or semi-structured format. Examples abound ranging from protein interaction networks (graphs) to phylogenetic datasets (trees), and from XML repositories of proteomic data (trees) to regulatory networks (graphs). The size and number of such data stores is growing rapidly.

Such data may arise directly out of experimental observations (e.g. PPI network complexes from mass spectrometry) or may be a convenient abstraction for housing relational information (e.g. Protein Data Bank). Other examples include mRNA measurements from microarray studies can be used to infer pairwise gene relations that imply co-expression of two genes. Regulatory relations between DNA binding proteins and genes can also be identified via various experimental technologies such as ChIP-chip, ChIP-seq, or DamID. Learning a biological network structure from experimental data that reflects the real world relations is a challenge in itself. Where data mining, in particular graph mining, can help is in the analysis of such structure data for the discovery of useful information. such as identification of common or useful substructures and detecting anomalous or unusual structures.

In this article we survey the use of graph mining for bioinformatics problems. This topic has been heavily researched over the last decade and we review the relevant material. We take a broad view of the term *graph mining* here. Since trees are simply connected acyclic graphs we include approaches that leverage tree mining algorithms as well. Additionally within the domain of graph mining there are approaches that focus on harvesting patterns from a single large graph or network and those that focus on extracting patterns from multiple graphs. We also cover other variants of graphs in our discussion including different tree variants, directed and bi-partite graphs.

The rest of this article is broadly divided into four sections. Section 2 discusses the use of tree mining algorithms for bioinformatics problems. For example, RNA secondary structures can be represented in the form of a tree. A forest of such RNA structure trees can be employed to characterize a newly sequenced novel RNA structure by identification of common topological patterns [93]. In particular we survey the role played by frequent tree mining algorithms, tree alignment, and statistical methods in this context.

In Section 3 we discuss algorithms that target the identification of frequent sub-patterns across multiple networks. For example in a recent study [53] it was shown how 39 co-expression networks of Budding Yeast can be analyzed

for coherent dense subgraphs across many of these networks. The discovered subgraphs then used to predict functionality of unknown genes. In particular we survey the role played by frequent graph mining algorithms and motif discovery algorithms in this context.

In Section 4 we discuss approaches that mine single and large biological networks for the identification of important subnetwork structures, such as identification of densely interacting communities from PPI networks or gene co-expression networks. In particular we discuss the role played by community discovery and graph clustering algorithms in the presence of uncertainty and noise in this context.

Finally in Section 5 we conclude this survey with a discussion of some open problems in the field.

2. Mining Trees

Trees are widely used to represent various biological structures like glycans, RNAs, and phylogenies.

Glycans are carbohydrate sugar chains attached to some lipids or proteins, and they are considered the third class of information-encoding biological macromolecules subsequent to DNA and proteins. The field of characterizing and studying is known as *glycomics*, akin to genomics and proteomics. Glycans play a critical role in many biological processes including embryonic development, cell to cell communication, coordination of immune functions, tumor progression, and protein regulations and interactions. Glycans are composed of monosaccharides (sugars) that are linked by glycosidic bonds. Unlike DNA and proteins which are simple strings of nucleotides and amino acids, monosaccharides may be linked to one or more other sugars, thereby forming a branched tree structure – they are often represented as rooted ordered labeled trees. In some cases, though rare, glycans may contain cycles due to rare cyclization of carbohydrate structures (e.g., cyclodextrins) [48]. There exist a number of representation schemes (KCF [5], LINUCS [13], GLYDE [87], GlycoCT [48], and GLYDE-II [83]) and database systems (CarbBank [1], SWEET-DB [75], KEGG/GLYCAN [45], EuroCarbDB [2], GlycoSuiteDB [26]) to store glycan data.

Ribonucleic acid (RNA) is a type of molecule that consists of a long chain of nucleotide units. RNA molecules play an important role in several key functionalities which include translation, splicing, gene regulation, and synthesis of proteins. As with all biomolecules, the function of RNAs is intimately related to their structure. The secondary structure of RNAs is a list of base

[1] http://bssv01.lancs.ac.uk/gig/pages/gag/carbbank.htm
[2] http://www.eurocarbdb.org/

pairs satisfying certain constraints. It is formed by folding the single-stranded RNA molecule back onto itself, and it provides a scaffold for the tertiary structure [82, 107]. The secondary structure is often modeled (with some approximations) as trees [11, 34, 35, 74, 93]. Since the exact experimental determination of RNA structure is difficult [33], scientists often employ computational methods for predicting the structure of various biological molecules. These methods provide a deeper understanding of RNAs structural repertoire, and thereby help in identifying new functional RNAs.

In *Phylogenetics*, trees are used as a fundamental data structure to represent and study evolutionary connections among different organisms as understood by ancestor–descendant relationships. The Tree of Life [3] is an example of such a tree illustrating the phylogeny of life on Earth that is based on the collective evidence from many different fields of biology and bioscience. The organisms over which a phylogenetic tree is induced are referred to as *taxa*, and they form the leaf nodes in the tree. The internal nodes denote the *speciation* and *duplication* events which result in *orthologs* and *paralogs*, respectively. Speciation is the origin of a new species capable of making a living in a new way from the species from which it arose. Paralogs are genes related by duplication within a genome. While traditional Phylogenetics relied on morphological data obtained by measuring and quantifying the phenotypic properties of representative organisms, more recent studies use gene or amino acid sequences encoding encoding proteins as the basis for classification. There exist a number of different approaches to construct these trees from input data [4] – distance matrix based methods, maximum parsimony, maximum likelihood, Bayesian inference etc. The trees produced by these methods can either be rooted or unrooted. Sometimes it is possible to force them to produce rooted trees by supplying an *outgroup*, which is an organism that is clearly less related to rest of the organisms. Such an outgroup is likely to be present near the root node. We now describe different techniques to analyze such tree structured biological data.

2.1 Frequent Subtree Mining

Frequent pattern mining is one of the fundamental data mining task that asks for a set of all substructures which appear more than a (user specified) threshold number of times in a given database. The subtree patterns obtained from tree databases are extremely useful in a variety of tasks such as structure prediction, identification of functional modules, consensus substructure discovery etc. We briefly describe some of these applications below.

[3] http://www.tolweb.org/tree/

[4] http://evolution.gs.washington.edu/phylip/software.html

The common techniques that are used to infer the phylogenies such as maximum parsimony [32] usually produce multiple trees for a given set of input sequences or genes. When the number of these output trees is too large to suggest meaningful evolutionary relations, Biologists use *consensus trees* or *supertrees* in order to summarize the output trees [77, 101]. One may also use such trees to infer common relations among trees produced from multiple different tree induction methods. Shasha and Zhang have studied the quality of consensus trees by extracting frequent *cousin pairs* from a set of phylogenetic trees modeled as rooted unordered trees [95]. A cousin pair defined as a pair of nodes which share the same ancestor node. The kinship in a cousin pair is captured via a distance measure that is measured using the depth of involved nodes. Given two parameters d and θ, their algorithm extracts all cousin pairs whose distance is at most d and whose frequency is at least θ. The discovered frequent pairs are also shown to be useful in discovering co-occurring patterns in multiple phylogenies, in evaluating the quality of consensus trees, and in finding kernel trees from a group of phylogenies.

The idea of frequent cousin pairs can be extended to more complex substructures, and they can be discovered by using traditional frequent subtree mining algorithms [117, 120]. From a biological standpoint, these agreement subtrees identify the set of species that are evolutionarily related according to a majority of trees under inspection. Zhang and Wang showed that these subtrees capture more important relationships when compared to consensus trees [120]. Hadzic *et al.* have applied similar methods on the 'Prions' database that describes protein instances stored for human Prion proteins [42].

Due to common evolutionary origins, there are often common substructures among multiple structurally similar RNAs. For instance, the occurrence of smaller snoRNA motifs within the larger hTR RNA structure, indicating a functional relation between these RNAs [79]. Uncovering such structural similarities is believed to help in discovering novel functional and evolutionary relationships among RNAs, which are not easily achieved by methods like sequence alignment [34]. Algorithms to extract common RNA substructures have been applied for the purpose of predicting RNA folding [69] and in functional studies of RNA processing mechanisms [93].

More recently, frequent subtree mining have been applied on glycan databases. Hashimoto *et al.* have developed an α-closed frequent subtree mining algorithm [46]. A frequent subtree S is considered α-closed unless support(S') $\geq$ max($\alpha \cdot$ support(T), $minsup$) for any supertree S' of S, where $0 \leq \alpha \leq 1$ and $minsup$ is the user defined support threshold. It mines maximal subtrees when α is set to 0 and closed subtrees when $\alpha = 1$. Instead of ranking the resulting subtrees based on their frequency, they rank them based on statistical hypothesis testing. This is because the frequencies of subtrees are easily biased by the frequencies of constituent monosaccharides. Based on their statistical

ranking method, they developed a glycan classification method that is similar to a well known linear soft margin SVMs [90]. Such a method essentially makes use of frequent subtrees obtained from a class of glycans in predicting whether or not a new glycan belongs to the given class.

2.2 Tree Alignment and Comparison

Comparison of two or more tree structures is a fundamental problem in many fields including RNA secondary structures comparison, syntactic pattern recognition, image clustering, genetics, chemical structure analysis, and Glycan structure analysis. The comparison among RNA secondary structures are known to be useful in identifying conserved structural motifs in folding process [93] and in constructing taxonomy trees [69]. The unordered tree comparisons can help in morphological problems arising in genetics – for example, in determining genetic diseases based on ancestry tree patterns [97].

Early research has focused on extending sequence matching algorithms to tree structures. The concepts related to longest common subsequence, shortest common supersequence, and string edit distance have been extended to largest common subtree (LCT) [1, 64, 118], smallest common supertree (SCS) [37, 41, 88, 110], and tree edit distance (TED) [12, 104, 119], respectively. In Phylogenetics, the longest common subtree problem is commonly referred to as Maximum Agreement Subtree (MAST) problem [36]. Biologists use MASTs to reconcile different evolutionary trees built over same taxa, and thereby to discover compatible relationships among those trees [63]. A number of efficient algorithms have been proposed for this purpose [31, 41, 64]. Aoki *et al.* studied the application of these techniques to index and query carbohydrate databases like KEGG [4].

Supertrees, on the other hand, can not only retain all or most of the information from the source trees but they can also find novel relationships which do not co-occur on any one source tree [88]. Supertrees in Phylogenetics can be built over source trees which share some but not necessarily all taxa. There are primarily two ways to build these supertrees. The first class of methods convert the topology of each source tree into a data matrix [85]. These matrices are then combined into a single large matrix, which is then used to construct the most parsimonious tree. When the given source trees are compatible, more direct methods can be used [25, 37]. In such a case, a backbone tree made up of taxa that common to given taxa is first constructed. By analyzing and thereby projecting each branch in backbone tree onto source trees, a combined supertree is constructed. The resulting supertrees are often referred to as *strict* since they do not conflict with any phylogenetic relationships in any source tree.

The tree edit distance between two trees refers to the number of minimum number of basic edit operations (relabel, insert, and delete) required to transform one tree into the other. This notion was first explored by Selkow [92], which was later generalized by Tai [104]. This conventional definition of edit distance has been extended to include more complex operations such as subtree insertions, subtree moves etc. [18, 17]. There has been a tremendous amount of work being done in developing fast algorithms to compute tree edit distance for both ordered and unordered trees. Most of the algorithms, similar to methods which compute string edit distance, follow dynamic programming based approaches. Bille has recently surveys several important algorithms that solve this problem [12]. These concepts have further been extended to RNA structures by taking their primary, secondary, and tertiary structures into account [40, 57].

Jiang *et al.* introduced the idea of *tree alignment* [58], which is in spirit similar to sequence alignment. An alignment between two trees is obtained by first inserting special nodes (labeled with spaces) into both trees such that the resulting trees have same structure. A cost model is defined over the set of opposing labels. The problem then is to find an optimal alignment which minimizes the sum of the costs of all opposing pairs [112]. Hochsmann *et al* designed a method for computing multiple alignments of RNA secondary structures, which was then used used to cluster RNA molecules purely based on their structure [50]. Bafna and Muthukrishnan presented a method to align a given RNA sequence with some unknown secondary structure to one with known sequence and structure. Such a method helps in RNA structure prediction in the case when the structure of a closely related sequence is known [9].

Glycan structure alignment techniques have been proposed by using traditional tree alignment algorithms and glycosidic linkage score matrices. These alignment techniques, just like popular sequence alignment methods, are useful when analyzing newly discovered glycans. Aoki *et al.* have proposed KCaM [5], an extension of popular Smith-Waterman sequence alignment technique [98], to perform exact and approximate glycan alignment. The approximate algorithm aligns monosaccharides while allowing gaps in the alignment, and the exact matching algorithm aligns linkages while disallowing any gaps, thus resulting in a stricter criterion for alignments. In a similar spirit, Aoki *et al.* have developed a glycan substitution matrix [2] to measure the similarity between monosaccharides, as in amino acid similarity represented by amino acid substitution matrices like BLOSUM [47]. Such a matrix can be used to discover those links that are positioned similarly, and thus potentially denote similar functionality. Thus, it is can be used to improve the alignment algorithms like KCaM to produce more biologically meaningful results. Kawano *et al.* have developed techniques to predict glycan structures from incomplete

or noisy data such as DNA microarray data by making use of knowledge about known glycan structures from KEGG GLYCAN database [62].

There is also an interesting notion of tree alignment, when the problem is discussed with respect to phylogenetic trees. While the traditional tree induction methods act upon sequence data to estimate the tree structure, tree alignment methods operate in reverse direction. More precisely, given a set of sequences from different species and a phylogenetic tree depicting the ancestral relationship among these species, compute an optimal alignment of the sequences by the means of constructing a minimum-cost evolutionary tree. Such methods are useful in determining the possible ancestral molecular sequences (which correspond to internal nodes in the tree) that gave rise to the extant sequences through a series of mutational events [56, 113].

2.3 Statistical Models

While analyzing glycan structures, unlike in phylogenies and RNA structures, it is often important to capture dependencies that are not bounded simply by the edges of the tree structure. In order to learn such patterns, a tree structured probabilistic model called as the Probabilistic Sibling-dependent Tree Markov Model (PSTMM) was developed [3, 108, 109]. It incorporates not only the dependency between a node and its its parent but also between a node and its eldest sibling. EM based learning algorithms were also proposed to learn parameters of the model. Hashimoto *et al.* have improved this for computational complexity by proposing ordered tree Markov model (OTMM) [44]. Instead of incorporating dependencies to both elder sibling and parent from each node, it uses only one dependency – where the eldest sibling depended only on the parent, and each younger sibling only depended on its older sibling. These methods have been applied to align multiple glycan trees, and thereby to detect biologically significant common subtrees in these alignments, where the trees are automatically classified into subtypes already known in glycobiology.

Ohtsubo and Marth showed that many motifs are involved in a variety of diseases including cancer i.e., these motifs act as *biomarkers* [81]. They also showed that the methods to predict characteristic glycan substructures (motifs) from a set of known glycans may be useful in predicting biomarkers of interest. Several research works have developed kernel methods for glycan biomarker classification and prediction. Hizukuri *et al.* developed a similarity measure known as *trimer kernel* for comparing glycan structures that takes the biological properties of involved glycans into account [49]. They have subsequently used this method in the framework of Support Vector Machines (SVMs) to extract characteristic functional units (motifs) specific to leukemia. This method was further extended by Koboyama *et al.* who developed a kernel that measures the similarity between two between two labeled trees by counting the

number of common q-length substrings known as *tree q-grams* [68]. Recently, Yamanishi *et al.* have developed a class of kernel functions which can be used for classifying glycans and detecting discriminative glycan motifs with SVMs [114]. The hierarchical model that they proposed handles the issue of large number of features required by the q-gram kernel. A kernel for each q was first developed, upon which another kernel was trained to extract the best feature from the best kernel.

3. Mining Graphs for the Discovery of Frequent Substructures

Graphs are important tools to model complex structures from various domains. Further characterization of these complex structures can be accomplished through the discovery of basic substructures that are frequently occurring. Identification of such repeating patterns might be useful for diverse biological applications such as classification of protein structural families, investigation of large and frequent sub-pathways in metabolic networks, and decomposition of Protein Protein Interaction (PPI) graphs into motifs. In this section, we focus on mining frequent subgraphs from biological networks. First, we look at various methods to identify subgraphs that occur frequently in a large collection of graphs. Next, we discuss substructures that occur significantly more often than expected by chance in a single and large graph, which are known as motifs. We cover different strategies for identification of such structures and their applications on diverse biological networks.

3.1 Frequent Subgraph Mining

Frequent subgraph mining (FSM) aims to find all (connected) frequent subgraphs from a graph database. More formally, given a set of graphs G, and a support threshold $minSup$, FSM finds all subgraphs (s_G) such that fraction of graphs in G of which s_G is a subgraph is greater than the $minSup$. There are two major challenges that are associated with FSM analysis: subgraph isomorphism and efficient enumeration of all frequent subgraphs. Subgraph isomorphism problem, which is an NP-complete problem, detects whether two given networks have the same structure. Therefore, time and space requirements for the existing FSM algorithms increase exponentially with the increasing pattern size and number of graphs. To design algorithms that scale to large biological graphs, techniques that simplify the problem by alternative graph modeling or graph summarization have been proposed. These algorithms are successfully utilized on diverse biological graphs for various purposes, including the identification of recurrently co-expressed gene groups and detection of frequently occurring subgraphs in a collection of metabolic pathways.

Koyuturk *et al.* developed a scalable algorithm for mining pathway substructures that are frequently encountered over different metabolic pathways [66]. A metabolic pathway is defined as a collection of metabolites M, enzymes Z, and reactions R. Each reaction $r \in R$, is associated with a set of enzymes ($Z(r) \in Z$) and a set of substrates and products which are metabolites. The algorithm aims to discover common motifs of enzyme interactions. Therefore, they re-model the metabolic pathways as directed graphs which emphasize enzyme interactions. In their representation, nodes represent enzymes, and a directed edge from an enzyme to another implies that the product of the first enzyme is consumed by a reaction catalyzed by the second. After constructing a collection of these graphs, they mine this collection to identify the maximal connected subgraphs that are contained in at least a pre-defined number of these graphs, where this number is determined by the support threshold. This model enforces unique node labeling to eliminate the subgraph isomorphism problem. This enforcement also enables the use of frequent itemset mining algorithms for the problem at hand by specifying edge-sets as the itemsets. In frequent itemset mining problem, each transaction is a collection of items, and the problem is to identify all frequent sets of items that occur in more than a specified number of these transactions. Koyuturk *et al,* reduced their problem into a frequent itemset mining problem by enforcing a connectivity constraint on edge-sets. They proposed an extension to a previously suggested frequent-itemset mining algorithm based on backtracking [38] which grows candidate subgraphs by only considering edges from a candidate edge set. Using their algorithm pathway graphs of 155 organisms collected from the KEGG database have been analyzed. They extracted considerably large sub-pathways that are frequent across these organism-specific pathway graphs. An example discovered sub-pathway of glutamate includes 4 nodes and 6 edges and it occurs in 45 of the 155 organisms. In a latter work, You *et al* applied SUBDUE system to obtain meaningful patterns from metabolic pathways [116]. SUBDUE is a system that identifies interesting and repetitive substructures based on graph compression and the minimum description length principles [51]. The best graphical pattern S that minimize the description length (MDL) of itself and that of the original input graph G when it is compressed with pattern S is identified with this system. First they identify the best pattern in G, which minimizes the MDL based criteria. Next, S is included into a hierarchy, where G is compressed with S. All such patterns in the input graph G are obtained, until no more compression is possible. The SUBDUE system is successfully applied on metabolic pathways to find unique and common patterns among a collection of pathways [116].

Another major application of FSM in biological domain is the identification of recurrent patterns from many gene co-expression networks. Gene co-expression networks are built on the basis of mRNA abundance measured by

microarray technologies. In a gene co-expression network, nodes represent genes, and two nodes are linked if the corresponding genes have significantly similar expression patterns over different microarray samples. Similarity between two genes is typically measured by the absolute value of the correlation coefficient between their expression profiles [52]. Next, based on a thresholding procedure, co-expression similarities are transformed into a measure of interaction strength. Different gene association networks can be constructed using different thresholding principles, i.e., hard or soft thresholding [52]. Although a gene co-expression network derived from a single microarray study can include many spurious edges, a recent study pointed out that genes co-expressed across multiple studies are more likely to be real and to correspond to functional groups [70]. Therefore, mining frequent gene groups across many gene co-expression networks has drawn recent attention. However, extant FSM algorithms do not scale to large gene co-expression graphs. In addition, as pointed by Hu *et al.*, frequency concept may not be enough to capture biologically interesting substructures. For this purpose, they proposed an algorithm, named CODENSE [53], that identifies frequent, coherent, and dense subgraphs across large collection of co-expression networks. According to their definition, all edges of a coherent subgraph frequently co-occur (and not co-occur) in the whole set of graphs. On the other hand, in a dense subgraph, the number of edges is close to the maximal possible number. Thus, coherent and dense structures better represent biological modules. Their algorithm starts with building a summary graph by eliminating infrequent edges from the input graphs. Another algorithm developed by the same group, MODES algorithm, is employed to extract dense subgraphs of the summary graph. For each of these dense summary subgraphs, edge occurrence profiles which is a binary matrix that indicates occurrence of dense summary graph edges in the original set of graphs are constructed. Using these profiles, a second-order graph is built to indicate the co-occurrence of edges across all graphs. In this representation, each edge is transformed into a node, and two nodes are connected if their corresponding edge occurrence profiles show high similarity. They shoved that coherent graphs across input graphs will be dense in the second-order graph. Therefore, at the final step of the CODENSE, dense subgraphs of the second-order graph are identified. CODENSE algorithm is scalable as it operates on two meta-graphs, namely summary graph and second order graph, instead of operating on individual networks. Dense patterns of these meta structures are identified, instead of patterns from individual graphs. It is also adjustable for exact or approximate pattern matching. CODENSE is applied on 39 co-expression networks of Budding Yeast organism to obtain functionally homogeneous gene clusters. These clusters are further employed in order to predict functionality of 169 unknown yeast genes. They showed that a significant portion of their predictions are supported by the literature [53].

CODENSE assumes that frequent subgraphs will be coherent across all graphs, on the other hand, it is possible to have subgraphs that are coherent only in a subset of these graphs. In order to take this fact into consideration, Huang *et al.* proposed an algorithm based on biclustering [55]. They start by identifying bi-cluster seeds from edge occurrence profiles. First, sub-matrices that are all 1s are identified from the edge co-occurrence matrix. Then, based on a Simulated Annealing methodology these initial structures are expanded. Connected components among these expanded seeds are identified and returned by their algorithm as recurring frequent subgraphs. They employed their algorithm on 65 co-expression datasets obtained from 65 different microarray studies. In a follow-up work conducted to identify frequently occurring gene subgraphs across many co-expression graphs, Yan *et al.* [115] studied a stepwise algorithm which constructs a neighbor association summary graph by clustering co-expression networks into groups. A neighbor association summary graph measures the association of two vertices based on their connections with their neighbors across input graphs. Two vertices that co-occur in many small frequent dense vertex sets have a high weight in the neighbor association graph. Once they build the neighbor association graph, they decompose it into (overlapping) dense subgraphs and then eliminate discovered dense subgraphs if their corresponding vertex-sets are not frequently dense enough. They named their algorithm NeMo for Network Module Mining. NeMo is applied on 105 human microarray datasets and recurrent co-expression clusters are identified. Functional homogeneity of these clusters are validated based on ChIP-chip data and conserved motif data [115].

For the automatic identification of common motifs in most any scientific molecular dataset, MotifMiner, a general and scalable toolkit has been proposed [23]. MotifMiner represents the information between a pair of nodes (atoms), A_i and A_j, as a mining bond. The mining bond $M(A_i, A_j)$ is a triplet of $< type(A_i), type(A_j), attr(A_i, A_j) >$ form. The information contained in $attr(A_i, A_j)$ vary depending on the resolution of the structure. As an example, if the structure is at the atomic level, $attr(A_i, A_j)$ can contain the distances between atoms A_i and A_j. This enables the flexibility to analyze several disparate domains, including protein, drug, and MD simulation datasets. Using mining bond definition, a k size structure is defined as $str_k = S, A_1, ..., A_k$, where A_i is the i^{th} atom and S is the set of mining bonds describing this structure. MotifMiner employs a Range pruning methodology to limit the search for viable strongly connected sub-structures and a Candidate pruning methodology to prune the search space of possible frequent structures. In addition, Recursive Fuzzy Hashing is used for rapid matching of structures while determining the frequency of occurrence. Distance Binning and Resolution principle is also proposed to work in conjunction with Recursive Fuzzy Hashing to handle noise in the input data. MotifMiner has been evaluated on various

datasets, including pharmaceutical data, tRNA data, protein data, molecular dynamics simulations [24]. In a follow-up study, Li *et al.* proposed several extensions, i.e., sliding resolution, handling boundary conditions, and enforcing local structure linkage, to the MotifMiner algorithm [72] in order to improve both the running time and the quality of the results. They also incorporated the domain constraints into the original MotifMiner algorithm for mining and aligning protein 3D structures. To evaluate the efficacy of the revised algorithm they used it to align the proteins Rad53 and Chk2, both of which contain FHA domain. FHA domains have very few conserved residues, which limits the use of sequence alignment algorithms for their alignment. The aligned result (depicted in Figure 18.1) is similar to structure-aided sequence alignment done manually [29], particularly at structurally similar regions. In a more recent work, a parallel implementation of this toolkit has been proposed [111]. The parallelized version demonstrate good speedup on real-world datasets.

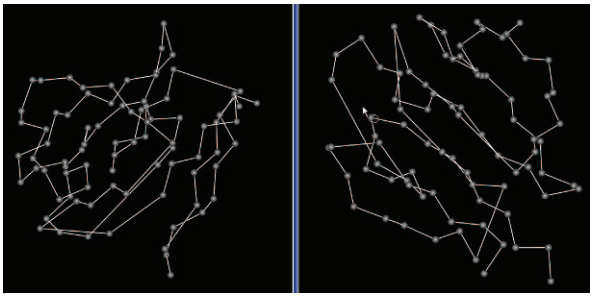

Figure 18.1. Structural alignment of two FHA domains. FHA1 of Rad53 (left) and FHA of Chk2 (right)

Jin *et al.* generalized the problem of frequent subgraph mining to mine frequent large-scale structures from graphs [59]. They developed a framework, Topological Structure Miner (TSMiner), that is based on a well-established mathematical concept known as topological minor. A topological minor of a given graph can be obtained by contracting the independent paths of one of its subgraphs into edges. Topological structures of a graph are derived from topological minors. Frequent subgraphs of a graph can be mined as a special case of frequent topological structures, but their framework is able to capture structures missed by standard algorithms. They proposed a scalable incremental algorithm to enumerate frequent topological structures. The concept of occurrence lists in order to efficiently count the support of a potential frequent topological structure is introduced. They employed this tool to search for potential protein-lipid binding sites in membrane proteins. Six membrane proteins, that are known to bind with cardiolipins (CL), are first represented in the form of graphs. In these graphs, amino acids represent nodes (20 different labels) and links exist between nodes if two amino acids are close enough to each other.

Two of the topological structures discovered with their toolkit are depicted in Figure 18.2. Such large structures cannot be obtained by using standard motif mining algorithms. As noted by the authors, the identified topological structures are mainly composed of polar (N, T, S), charged (K), and aromatic (W) residues, which is in agreement with biophysics literature.

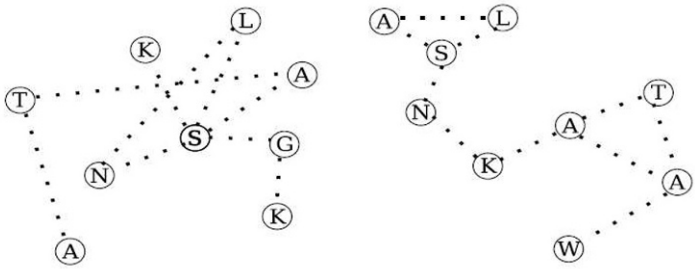

Figure 18.2. Frequent Topological Structures Discovered by TSMiner

3.2 Motif Discovery in Biological Networks

In addition to subgraphs that are frequent across many networks, substructures that are repeated frequently within a single and large network can be useful for knowledge discovery. A motif of a graph refers to a substructure, which is repeated considerably inside the graph. There are two main approaches, frequency-based and statistical, to determine the significance of this repetition. The frequency-based approach considers a subgraph as a motif if it is occurring more than a threshold number of times. On the other hand, statistical approach labels a subgraph as motif if it is occurring more than the expected number of times with respect to random networks. Network motifs can be particularly effective in understanding the modularity and the global structure of biological networks. For example in the case of PPI networks, motifs can be useful for the identification of protein complexes and other protein groupings that are related to the mechanics of the living organism. In the case of regulatory networks, motifs enable understanding gene regulation mechanisms and it also enables researchers to develop models and experiments to understand these mechanics.

Milo *et al.* is the first to define network motifs and find them in networks from biochemistry, neurobiology, ecology, and engineering [78]. They defined network motifs as *patterns of interconnections occurring in complex networks at numbers that are significantly higher than those in randomized networks*. Their analysis revealed some common (and diverse) motifs across fields. As an example, they shoved that the directed triangle motif, known as the feed-forward loop, exists in both transcription-regulatory and neural networks, whereas four-node feedback loops only emerge in electric circuits but

not in biological systems. To identify such motifs, Milo *et al.* exhaustively enumerated all subgraphs of n nodes in the studied networks, where n is limited to 3 and 4. They then generated random networks while keeping the number of nodes, links and the degree distribution unchanged. Subgraphs of these random networks are counted and these counts are used to determine motifs. As an alternative to exact counting, in a follow-up work they proposed a *sampling method for subgraph counting* [61]. Instead of enumerating subgraphs exhaustively, subgraphs are sampled to estimate their relative frequency. The method starts by picking a random edge from the network and then expanding the corresponding subgraph iteratively by picking random neighboring edges. At each iteration, a list of all candidate edges are generated for the next random pick. The subgraph is expanded until it reaches a pre-defined size. Although being an extension over the exhaustive search, this algorithm is also limited to finding small-size motifs. In the transcription network of E. coli, subgraph samples of sizes 3 to 8 have been reported. Higher order motifs composed of five and six nodes in this network are tabulated in their study [61].

Protein-protein interaction networks accumulate pairwise or group-wise physical interactions of proteins into a network structure. Motifs of these networks can be utilized to characterize and better understand the group-level relations. For identification of large size motifs in Protein-Protein Interaction (PPI) networks, a scalable algorithm, NEtwork MOtif FINDER [19] has been proposed as an extension to subgraph mining algorithms. This algorithm is based on formation of frequent trees of varying size from 2 to k, which are then used to partition the graph into a set of graphs such that each graph embeds a size-k tree. In the next step, frequent size-k graphs are generated by performing graph join operations. Frequency of these size-k graphs can be counted in randomized networks. NEMOFINDER describes frequent subgraphs that are also unique as *Network Motifs*. Uniqueness of a subgraph is determined by the number of times a subgraph is more frequent in the real graph than randomized graphs. Existing Apriori-based algorithm are not able to capture interesting network motifs that are repeated and unique. Uniqueness of these size-k graphs are calculated based on their number of occurrences in real input graph and the randomized graphs. They build their algorithm as an extension to the SPIN [54] algorithm with the possibility of overlapping subgraphs. The input to the NEMOFINDER algorithm is a PPI network, and user defined thresholds for frequency, uniqueness, and maximal network size. The algorithm outputs Network Motifs that are frequent and unique with respect to the defined thresholds. Employing their algorithm on the PPI network of budding yeast, they discovered motifs up to size 12. They later proposed an extension to the NEMOFINDER, named LaMoFinder, which takes into consideration labels of nodes [20]. While applying LaMoFinder to discover PPI network motifs, they used Gene Ontology terms as node labels [20]. They first mine

an unannotated network for motifs. Next, motifs are labeled with Gene Ontology functions. Their analysis showed that by incorporating labels they are not able to capture only the topological shapes but also biological context of motifs. Labeled motifs extracted from a real world PPI network are employed for protein function prediction.

In a more recent work, Grochow and Kellis [39] proposed an algorithm to avoid the limitations of exact counting and subgraph sampling based motif mining algorithms. Their algorithm works by exhaustively searching for instances of a single query graph in a network. They proposed a motif-centric alternative to existing methods which is based on an improved isomorphism test, i.e., symmetry breaking. The algorithm identifies all instances of a query graph H, given a target network G. They extended isomorphism test based on the most constrained neighbor concept. They defined the most constrained neighbor of the already-mapped nodes which is the least possible nodes to be mapped to. They also introduced and enforced several symmetry-braking conditions, to make sure that there is a unique map from the query graph H to each instance of H in G. They utilized their algorithm to find motifs in two biological networks: PPI network of S. cerevisiae and Transcriptional network of S. cerevisiae. The former is composed of 1379 nodes and 2473 edges, where motifs of 15 and 20 nodes can be identified with the proposed algorithm. From the latter one, which has 685 nodes and 1052 edges, a 29-node motif that corresponds to the cellular transcription machinery has been identified. In addition to being scalable for finding larger motifs, this algorithm also enables exploring motif clustering and querying a particular subgraph. Moreover, the algorithm is very easy to parallelize by counting each subgraph on a separate processor.

4. Mining Graphs for the Discovery of Modules

Different forms of real-life associations between biological entities have been detected by various technologies and these associations have been accumulated in the public databases in the form of complex networks. Understanding these complex structures often require breaking them into small components and identifying the interactions between these components. These components are composed of nodes which are more relevant to each other than with outsiders and they are commonly referred as communities or modules. Decomposition of a given graph into its modules can also be very effective in the analysis of biological networks. Some biological networks are naturally decomposed into such components, which are commonly referred as modular networks. Some examples of biological modules are transcriptional modules, protein complexes, gene functional groups, and signaling pathways.

The most well-known biological modular networks is the Protein-Protein Interaction(PPI) Network. The number and coverage of public databases that collect experimental data on protein physical bindings of diverse organisms have been increasing with the advancements in high-throughput techniques. Although there is no established standard database of PPIs today, there have been efforts to integrate existing interactions in publicly available databases. As of today, Human Protein Reference Database (HPRD) footnote`http://www.hprd.org`) includes 34,624 Protein-protein interactions between Human proteins that are derived from a number of platforms such as Mass Spectrometric Analysis, Yeast two-hybrid based protein-protein interaction, and Co-immunoprecipitation and mass spectrometry-based protein-protein interaction. Similarly, another freely accessible database BIOGRID [100] includes more than 238,634 raw interactions from various organisms including Saccharomyces cerevisiae, Caenorhabditis elegans, Drosophil melanogaster and Homo sapiens. These large collections of protein interactions are naturally represented in the form of networks to facilitate the process of knowledge discovery. Modular nature of these networks has been investigated by different algorithms and the identified modules have been utilized for a better characterization of the unknown proteins.

Gene co-expression networks are another example of biological networks that exhibit modular structure [15, 102]. In these network structures, nodes represent genes and edges between nodes refer to genes that are expressed similarly over studied conditions. Gene groups that indicate a similar expression pattern can be defined as a gene module, where a functionality between the elements of this module is likely to be shared [91, 102]. Another modular biological network that have been excessively studied is the Regulatory networks. They model activation (or suppression) of a gene by specific DNA binding proteins in the form of a directed graph. Modules that can be deduced from regulatory networks correspond to a set of co-regulated genes as well as their common regulators. Given all these application areas, effective identification of modules from diverse biological networks has great potential for a better understanding of studied organisms.

In this section we discuss different methodologies that are proposed for the detection of network modules or communities in biological graphs. Here, a community can be defined as a densely connected group of nodes, where only a few connections exist between different communities [80]. First, we look at algorithms that extract community structures from networks. Next, we discuss clustering algorithms that have been proposed to decompose the whole structure into subgroups, where similarity within group elements is maximized, and between groups is minimized.

4.1 Extracting Communities

In the analysis of PPI networks, of particular interest to many scientists is to study protein interaction networks to isolate densely interacting regions, also known as communities, since they are presumed to be protein complexes or functional modules. A protein complex can be defined as a set of proteins that bind to each other in order to accomplish a cellular level task. Identification of these structures is useful to understand cell functioning, to predict functionality of unknown proteins. The interest in their identification is motivated by the fact that proteins heavily interacting within themselves, usually participate into the same biological processes. Thus, discovery of dense subgraphs from PPI networks is recognized as an important task for the identification of protein complexes. Based on this underlying principle, a set of algorithms that employ local dense regions of PPI networks to discover putative complexes have been proposed.

Bader et al [8] proposed a three-step algorithm; Molecular COmplex DEtection (MCODE) to identify clusters of proteins that are heavily interacting. MCODE starts with weighting each node of the network based on the density of its local neighborhood. Next, nodes with high weights are assigned as seeds and starting from these seed nodes initial clusters are obtained by iteratively including neighboring nodes to the cluster. Finally an optional third step is proposed to filter proteins according to a connectivity criteria. They evaluated MCODE on an integrated dataset of Budding Yeast that is composed of 9088 protein-protein interactions among 4379 proteins from the MIPS, YPD, and PreBIND databases. They predicted 166 complexes from this network. 52 of these complexes matched with known protein complexes in the MIPS database. MCODE bases on the observation that proteins share functions with their immediate neighbors. In a more recent work, Chua et al utilized another observation based on level-2 interactions in PPI networks [22]. They derived a topological weighting schema, namely the Functional Similarity Weight (FS-Weight) that enables weighting both direct and indirect (i.e., 'level-2') interactions. FS-Weight makes use of estimated reliability of each interaction to reduce the impact of noise. The reliability of each experimental source is estimated by the fraction of unique interactions in which at least one level-4 Gene Ontology term is shared. FS-Weight also favors two proteins that share many common neighbors from a reliable source. Number of non-common neighbors are also included into the calculation in order to reduce potential false positive inferences. Based on FS-weights, the studied PPI network is expanded with 'level-2' interactions and filtered by eliminating interactions with small FS-weights. After this preprocessing step, they identify cliques in the modified PPI network and iteratively merged cliques to form larger subgraphs that are still dense. More recently, Li et al [73] proposed an algorithm named DE-

CAFF (Dense Neighborhood Extraction using Connectivity and conFidence measures) which employs the Hub Removals algorithm [86]. DECAFF initially identifies local dense neighborhoods of each protein by iteratively removing nodes with low degrees from the local neighborhoods. These local cliques are merged with the dense subgraphs detected by the Hub Removal algorithm [86] based on a Neighborhood Affinity criteria. Neighborhood Affinity of two subgraphs is calculated based on their size and the number of their common neighbors. Finally DECAFF improves the quality of final clusters by removing subgraphs with low reliability scores. The reliability of a subgraph is defined as the average reliability of all interactions of that subgraph, where interaction reliability is deduced from functional relevance of its two interacting proteins.

In addition to PPI networks, scientists are also interested in identifying community structures from gene co-expression networks. Expression profiles obtained through microarray studies can be transformed into gene co-expression networks, where nodes represent genes and two nodes are linked if the corresponding genes behave significantly similar across different samples (i.e., co-expression). Scientists are particularly interested in the problem of identifying gene subnetworks that have similar expression patterns under different conditions [103] since they have been theorized to have the same cellular function [30]. To find gene groups that have similar expression patterns, Hartuv and Shamir proposed an algorithm that recursively splits the weighted co-expression graph into its highly connected components [43]. A highly connected component is defined as a subnetwork which includes at least two nodes, i.e., $n > 1$, and which can only be disconnected after the removal of more than $n/2$ edges. Their algorithm, namely the *Highly Connected Subgraphs*(HCS), at each iteration splits the network into subgraphs until a highly connected component is identified. Shamir and Sharan [94] proposed an extension of the HCS algorithm, CLICK - CLuster Identification via Connectivity Kernels. In each step of their algorithm, a minimum cut of the input graph is computed, which outputs two subgraphs. Subgraphs which satisfied certain criterion are labeled as kernels. Each kernel is attributed with a fingerprint similarity that is calculated based on its elements. After all the kernels are identified, nodes that are not part of any kernels are further analyzed and the ones that are similar to any of the kernels are included into the kernel and the kernel's fingerprint is re-calculated - adoption step in the algorithm. Next, kernels that are similar enough are merged and the adoption operation is repeated. Adoption and kernel merging steps are repeated until there are no more changes in the kernel structures. Final kernels are outputted as gene clusters obtained by the CLICK algorithm. They have shown that their algorithm outperform existing clustering algorithms when applied on various gene expression datasets,

originating from various studies, such as the yeast cell cycle dataset, or the response of human fibroblasts to serum.

Regulatory modules can be inferred from diverse datasets including ChIP-chip, motif, and gene expression datasets. A regulatory module is composed of a set of genes that are co-regulated by a common set of regulators. In order to identify such modules from ChIP-chip data and gene expression profiles, GRAM algorithm is proposed [10]. A set of genes that are bind with the same regulator set is obtained from the ChIP-chip binding p-values with an exhaustive search. Subsequently, a subset of this set that are similarly expressed is selected to serve as a seed. Then, the algorithm identifies genes that are similarly expressed with the seed genes and that are connected to the same set of transcription factors based on a relaxed binding criteria. Lemmens *et al.* improved the GRAM algorithm by incorporating motif data as an additional source [71]. In the seed discovery step, they discover seeds composed of genes that are co-expressed (deduced from mRNA measurements), that bind to the same regulators (deduced from ChIP-chip data), and that have the same motifs in their intergenic regions (deduced from Motif data). they employed an Apriori-like algorithm in order to identify such seeds. And a p-value is assigned to asses the quality of each seed. In the second seed extension step, gene content of the seeds are extended. For this purpose, each gene is ranked according to their correlation with the mean expression profile of the seed genes, and the ones that are similar enough (according to a cut-off) are included into the module. They employed their algorithm for the discovery of Budding Yeast regulatory modules by integrating ChIP-chip, motif, and gene expression datasets.

4.2 Clustering

Clustering algorithms can also be effective in identifying the modules of biological networks. In contrast to community discovery approaches, clustering (or graph partitioning) decompose the whole network structure into groups. A clustering algorithm locates every node of the graph into a community or a module.

To elucidate gene functions at a global scale, clustering of gene co-expression networks have been investigated. Since genes that are on the same pathways or belong to the same functional complexes are often co-regulated, they often exhibit similar expression patterns under diverse conditions. Thus, identifying and studying groups of highly-interacting genes in co-expression networks is an important step towards characterizing genes at a global scale. For this purpose, a variety of existing graph partitioning algorithms can be leveraged. Spectral methods that target weighted cuts [96] form an important class of such algorithms. Multi-level graph partitioning algorithms such as Metis [60] and Graclus[27] are well known to scale well for large networks.

Divisive/agglomerative approaches have also been popular in network analysis [80], but they are expensive and do not scale well [16]. Markov Clustering (MCL) [28], a graph clustering algorithm based on (stochastic) flow simulation, has proved to be highly effective at clustering biological networks [14]. A variant of this algorithm known as MLR-MCL [89] have been proposed recently to address the scalability of MCL algorithm.

In addition to these diverse graph partitioning algorithms, other classical clustering algorithms have also been employed – e.g., the hierarchical clustering [99], the k-means clustering [76], and the self-organizing maps [65]. Besides the application of standard clustering algorithms, clustering algorithms that are more suitable for the specific task have been studied. Among these are the biclustering algorithms which identify a group of genes that behave similarly only for a subset of all conditions. Given a gene expression matrix of samples and genes, biclustering algorithms perform clustering in two dimensions simultaneously [21]. Statistically significant sub-matrices of a subset of genes and a subset of samples are the identified biclusters. Cheng and Church proposed a greedy approach in order to find maximal sized biclusters that satisfy a certain condition on the residue scores [21]. Their algorithm identifies each biclusters separately by iteratively removing rows and columns until the mean squared residue score for the sub-matrix (an assessment for the quality of bi-cluster) is smaller than a threshold and by iteratively adding rows and columns while the quality assessment score does not exceed threshold. Each run of the algorithm identifies a sub-matrix (bi-cluster) separately, and the next bi-cluster is identified after the found sub-matrix is masked by randomization. using this algorithm, they identified biclusters from gene expression datasets of Human and Yeast. Later, Koyuturk *et al* proposed a work which associates statistical significance to the extracted biclusters. To discover binary biclusters from a quantized gene expression matrix, they formulate this problem as an optimization problem based on the statistical significance objective. Fast heuristics are proposed so solve this optimization problem in a scalable manner. The algorithm is tested on quantized breast tumor gene expression matrix [67]. Tanay *et al.* converted bi-clustering problem into a graph theory problem using bi-partite modeling [106]. Initially the expression data is converted into a bi-partite of genes and samples. More formally a graph $G(V, S, E)$ is constructed where V is set of genes, S is set of conditions, and there exists and edge between v and s, $(v, s) \in E$ if, g is expressionally responsive in sample s. This modeling reduces the biclustering problem into the problem of finding the densest subgraphs in G. Since the identification of heaviest bi-clique is an NP-complete problem, authors restricted the search space by assuming a degree bound on one side of the bipartite graph. Later Tanay applied SAMBA algorithm on the gene expression dataset of 96 human tissue samples [105]. In that work, they compared their work against, Cheng

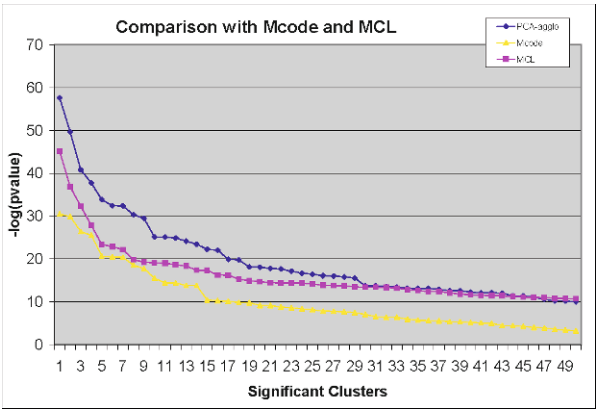

Figure 18.3. Benefits of Ensemble Strategy for Community Discovery in PPI networks in comparison to community detection algorithm MCODE and clustering algorithm MCL. The Y-axis represents -log(p-value).

and Church's algorithm [21] and observed that biclusters from SAMBA are better in terms of their statistical significance.

An ensemble clustering algorithm is also studied on biological networks to generate a more robust clustering compared to individual clustering algorithms [6]. Cluster ensembles can be defined as a mapping from a set of clusterings generated by a variety of sources into a single consensus clustering arrangement. Asur *et al.* proposed an ensemble clustering for the PPI decomposition problem. First different topological weighting schemes are proposed to generate different views of the unweighted PPI network. Next, these different views are clustered with different algorithms to obtain a set of base clusterings of the network. These clusterings are integrated into a Cluster Membership Matrix which is reduced in size to eliminate redundancy and to scale the consensus determination problem based on PCA. Subsequently standard hierarchical clustering algorithms are utilized for computing the consensus clustering (recursive bisections (PCA-rbr) and agglomerative clustering (PCA-agglo)). When compared with existing community detection and clustering algorithms, they observed that their algorithm is able to produce topologically and biologically more significant clusters (as shown in Figure 18.3). The Y-axis represents distribution of Gene Ontology enrichment p-values. Smaller p-values represent more significantly enriched groups with a particular Gene Ontology term.

In addition to biclustering and ensemble clustering strategies, scientists also studied soft clustering algorithms for biological networks, which enables assigning multiple-cluster membership to multi-faceted biological entities. To

enable multiple cluster membership for proteins while identifying PPI clusters, Asur et al [6] proposed a soft ensemble clustering technique that is a step further from their PCA based consensus clustering. This adapted algorithm, after obtaining the initial consensus clustering, iteratively calculates the strength of each protein's membership to each consensus cluster based on shortest path distances. Proteins that have high propensity towards multiple membership are then assigned to their alternate clusters. To test the efficacy of this soft clustering algorithm, the compared their algorithm with the original ensemble clustering. As can be seen in Figure 18.4, they observed that, allowing multiple membership to proteins, improves the overall accuracy of the clustering, as evident from the smaller p-values of GO enrichment analysis.

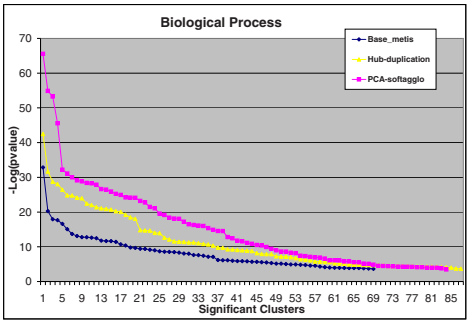

Figure 18.4. Soft Ensemble Clustering improves the quality of extracted clusters. The Y-axis represents -log(p-value).

A soft bi-clustering algorithm (MF-PINCoC), an extension to the algorithm PINCoC, has been proposed to identify overlapping dense subgraphs by using a local search technique has been proposed recently [84]. The PINCoC algorithm applies a greedy search strategy in order to find the local optimal sub-matrices in terms of a quality function. More recently, Avogadri *et al.* proposed an ensemble fuzzy clustering for decomposing gene expression datasets into its overlapping clusters [7]. They first generate multiple views of the data by using random projections. A random projection maps data from a high-dimensional space to a lower dimensional space. On these views, they applied fuzzy k-means algorithm and these fuzzy clustering arrangements are combined into a similarity matrix. They again employed fuzzy k-means on this similarity matrix to identify fuzzy consensus clustering [7]. This algorithm is applied on four different microarray datasets and compared against different ensemble strategies.

5. Discussion

In this article we surveyed the principal results in the field of graph mining that relate to the application domain of bioinformatics. We examined these results along three directions: i) from the perspective of mining tree-structured data; ii) from the perspective of mining multiple graphs or networks; and iii)

from the perspective mining of mining a single (large) network in the presence of noise and uncertainty.

Both data mining and the field of bioinformatics are young and vibrant and thus there are ample opportunities for interesting lines of future research at their intersection. Sticking to the theme of this article – graph mining in bioinformatics – below we list several such opportunities. This list is by no means a comprehensive list but highlight some of the potential opportunities researchers may avail of.

- Scalable algorithms for analyzing time varying networks: A large majority of the work to date in this field has focused on the analysis of static networks. While there have been some recent efforts to analyze dynamic biological networks, research in this arena is at its infancy. With anticipated advances in technology where much more temporal data is likely to become available temporal analysis of such networks is likely to be an important arena of future research. Underpinning this effort, given the size and dynamics of the data involved are the need to develop scalable algorithms for processing and analyzing such data.

- Discovering anomalous structures in graph data: Again while most of the work to date has focused on the discovery of frequent or modular structure within such data – the discovery of anomalous substructures often has a crucial role to play in such domains. Defining what constitutes an anomaly, how to compute it efficiently while leveraging the ambient knowledge in the domain in question are some of the challenges to be addressed.

- Integrating data from multiple, possibly conflicting sources: A fundamental challenge in bioinformatics in general is that of data integration. Data is available in many formats and often times are in conflict. For example protein interaction data produced by various experimental methods (mass spectrometry, Yeast2Hybrid, in-silico) are often in conflict. Research into methods that are capable of resolving such conflicts while still discovering useful patterns are needed.

- Incorporating domain information: It has been our observation that often we as data mining researchers tend to under-utilize available domain information. This may arise out of ignorance (the field of bioinformatics is very vast) or simply omitted from the training phase as a means to confirm the utility of the proposed methods (to maintain the sanctity of the validation procedure). We believe a fresh look at how domain knowledge can be embedded in existing approaches and better validation methodologies in close conjunction with domain experts must be looked into.

- Uncertainty-aware and noise-tolerant methods: While this has certainly been an active area of research in the bioinformatics community in general, and in the field of graph mining in bioinformatics in particular, there are still many open problems here. Incorporating uncertainty is necessarily a domain-dependent issue and probabilistic approaches offer exciting possibilities. Additionally leveraging topological, relational and other semantic characteristics of the data effectively is an interesting topic for future research. A related challenge here is to model trust and provenance related information.

- Ranking and summarizing patterns harvested: While ranking and summarizing patterns has been the subject of much research in the data mining and network science community the role of such methods in bioinformatics has been much less researched. We expect this to be a very important and active area of research especially since often times evaluating and validating patterns discovered can be an expensive and time consuming process. In this context research into ranking algorithms for bioinformatics that leverage domain knowledge and mechanisms for summarizing patterns harvested is an exciting opportunity for future research.

References

[1] Akutsu, T. (1992). An RNC algorithm for finding a largest common subtree of two trees. *IEICE Transactions on Information and Systems*, 75(1):95–101.

[2] Aoki, K., Mamitsuka, H., Akutsu, T., and Kanehisa, M. (2005). A score matrix to reveal the hidden links in glycans. *Bioinformatics*, 21(8):1457–1463.

[3] Aoki, K., Ueda, N., Yamaguchi, A., Kanehisa, M., Akutsu, T., and Mamitsuka, H. (2004a). Application of a new probabilistic model for recognizing complex patterns in glycans.

[4] Aoki, K., Yamaguchi, A., Okuno, Y., Akutsu, T., Ueda, N., Kanehisa, M., and Mamitsuka, H. (2003). Efficient tree-matching methods for accurate carbohydrate database queries. *Genome Informatics Sl*, pages 134–143.

[5] Aoki, K., Yamaguchi, A., Ueda, N., Akutsu, T., Mamitsuka, H., Goto, S., and Kanehisa, M. (2004b). KCaM (KEGG Carbohydrate Matcher): a software tool for analyzing the structures of carbohydrate sugar chains. *Nucleic acids research*, 32(Web Server Issue):W267.

[6] Asur, S., Ucar, D., and Parthasarathy, S. (2007). An ensemble framework for clustering protein protein interaction networks. *Bioinformatics*, 23(13):i29.

[7] Avogadri, R. and Valentini, G. (2009). Fuzzy ensemble clustering based on random projections for DNA microarray data analysis. *Artificial Intelligence in Medicine*, 45(2-3):173–183.

[8] Bader, G. and Hogue, C. (2003). An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinfomatics*, 4:2.

[9] Bafna, V., Muthukrishnan, S., and Ravi, R. (1995). Computing similarity between RNA strings. *In Combinatorial Pattern Matching (CPM), volume 937 of LNCS*.

[10] Bar-Joseph, Z., Gerber, G., Lee, T., Rinaldi, N., Yoo, J., Robert, F., Gordon, D., Fraenkel, E., Jaakkola, T., Young, R., et al. (2003). Computational discovery of gene modules and regulatory networks. *Nature Biotechnology*, 21(11):1337–1342.

[11] Benedetti, G. and Morosetti, S. (1996). A graph-topological approach to recognition of pattern and similarity in RNA secondary structures. *Biophysical chemistry*, 59(1-2):179–184.

[12] Bille, P. (2005). A survey on tree edit distance and related problems. *Theoretical computer science*, 337(1-3):217–239.

[13] Bohne-Lang, A., Lang, E., Førster, T., and von der Lieth, C. (2001). LINUCS: linear notation for unique description of carbohydrate sequences. *Carbohydrate research*, 336(1):1–11.

[14] Brohee, S. and van Helden, J. (2006). Evaluation of clustering algorithms for protein-protein interaction networks. *BMC bioinformatics*, 7(1):488.

[15] Butte, A. and Kohane, I. (2000). Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. In *Pac Symp Biocomput*, volume 5, pages 418–429.

[16] Chakrabarti, D. and Faloutsos, C. (2006). Graph mining: Laws, generators, and algorithms. *ACM Computing Surveys (CSUR)*, 38(1).

[17] Chawathe, S. and Garcia-Molina, H. (1997). Meaningful change detection in structured data. *ACM SIGMOD Record*, 26(2):26–37.

[18] Chawathe, S., Rajaraman, A., Garcia-Molina, H., and Widom, J. (1996). Change detection in hierarchically structured information. In *Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, pages 493–504. ACM New York, NY, USA.

[19] Chen, J., Hsu, W., Lee, M., and Ng, S. (2006). NeMoFinder: Dissecting genome-wide protein-protein interactions with meso-scale network motifs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 106–115. ACM New York, NY, USA.

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

[21] Cheng, Y. and Church, G. (2000). Biclustering of expression data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology table of contents*, pages 93–103. AAAI Press.

[22] Chua, H., Ning, K., Sung, W., Leong, H., and Wong, L. (2007). Using indirect protein-protein interactions for protein complex prediction. In *Computational Systems Bioinformatics: Proceedings of the CSB 2007 Conference*, page 97. Imperial College Press.

[23] Coatney, M. and Parthasarathy, S. (2005a). MotifMiner: Efficient discovery of common substructures in biochemical molecules. *Knowledge and Information Systems*, 7(2):202–223.

[24] Coatney, M. and Parthasarathy, S. (2005b). Motifminer: Efficient discovery of common substructures in biochemical molecules. *Knowl. Inf. Syst.*, 7(2):202–223.

[25] Constantinescu, M. and Sankoff, D. (1995). An efficient algorithm for supertrees. *Journal of Classification*, 12(1):101–112.

[26] Cooper, C., Harrison, M., Wilkins, M., and Packer, N. (2001). GlycoSuiteDB: a new curated relational database of glycoprotein glycan structures and their biological sources. *Nucleic Acids Research*, 29(1):332.

[27] Dhillon, I., Guan, Y., and Kulis, B. (2005). A fast kernel-based multilevel algorithm for graph clustering. *Proceedings of the 11th ACM SIGKDD*, pages 629–634.

[28] Dongen, S. (2000). *Graph clustering by flow simulation*. PhD thesis, PhD Thesis, University of Utrecht, The Netherlands.

[29] Durocher, D., Taylor, I., Sarbassova, D., Haire, L., Westcott, S., Jackson, S., Smerdon, S., and Yaffe, M. (2000). The molecular basis of FHA domain: phosphopeptide binding specificity and implications for phosphodependent signaling mechanisms. *Molecular Cell*, 6(5):1169–1182.

[30] Eisen, M., Spellman, P., Brown, P., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868.

[31] Farach, M. and Thorup, M. (1994). Fast comparison of evolutionary trees. In *Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 481–488. Society for Industrial and Applied Mathematics Philadelphia, PA, USA.

[32] Fitch, W. (1971). Toward defining the course of evolution: minimum change for a specific tree topology. *Systematic zoology*, 20(4):406–416.

[33] Fürtig, B., Richter, C., Wohnert, J., and Schwalbe, H. (2003). NMR spectroscopy of RNA. *ChemBioChem*, 4(10):936–962.

[34] Gan, H., Pasquali, S., and Schlick, T. (2003). Exploring the repertoire of RNA secondary motifs using graph theory; implications for RNA design. *Nucleic acids research*, 31(11):2926.

[35] Gardner, P. and Giegerich, R. (2004). A comprehensive comparison of comparative RNA structure prediction approaches. *BMC bioinformatics*, 5(1):140.

[36] Gordon, A. (1979). A measure of the agreement between rankings. *Biometrika*, 66(1):7–15.

[37] Gordon, A. (1986). Consensus supertrees: the synthesis of rooted trees containing overlapping sets of labeled leaves. *Journal of Classification*, 3(2):335–348.

[38] Gouda, K. and Zaki, M. (2001). Efficiently mining maximal frequent itemsets. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 163–170.

[39] Grochow, J. and Kellis, M. (2007). Network motif discovery using subgraph enumeration and symmetry-breaking. *Lecture Notes in Computer Science*, 4453:92.

[40] Guignon, V., Chauve, C., and Hamel, S. (2005). An edit distance between RNA stem-loops. *Lecture notes in computer science*, 3772:333.

[41] Gupta, A. and Nishimura, N. (1998). Finding largest subtrees and smallest supertrees. *Algorithmica*, 21(2):183–210.

[42] Hadzic, F., Dillon, T., Sidhu, A., Chang, E., and Tan, H. (2006). Mining substructures in protein data. In *IEEE ICDM 2006 Workshop on Data Mining in Bioinformatics (DMB 2006)*, pages 18–22.

[43] Hartuv, E. and Shamir, R. (2000). A clustering algorithm based on graph connectivity. *Information processing letters*, 76(4-6):175–181.

[44] Hashimoto, K., Aoki-Kinoshita, K., Ueda, N., Kanehisa, M., and Mamitsuka, H. (2006a). A new efficient probabilistic model for mining labeled ordered trees. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 177–186.

[45] Hashimoto, K., Goto, S., Kawano, S., Aoki-Kinoshita, K., Ueda, N., Hamajima, M., Kawasaki, T., and Kanehisa, M. (2006b). KEGG as a glycome informatics resource. *Glycobiology*, 16(5):63–70.

[46] Hashimoto, K., Takigawa, I., Shiga, M., Kanehisa, M., and Mamitsuka, H. (2008). Mining significant tree patterns in carbohydrate sugar chains. *Bioinformatics*, 24(16):i167.

[47] Henikoff, S. and Henikoff, J. (1992). Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919.

[48] Herget, S., Ranzinger, R., Maass, K., and Lieth, C. (2008). GlycoCT: a unifying sequence format for carbohydrates. *Carbohydrate Research*, 343(12):2162–2171.

[49] Hizukuri, Y., Yamanishi, Y., Nakamura, O., Yagi, F., Goto, S., and Kanehisa, M. (2005). Extraction of leukemia specific glycan motifs in humans by computational glycomics. *Carbohydrate research*, 340(14):2270–2278.

[50] Hɵchsmann, M., Voss, B., and Giegerich, R. (2004). Pure multiple RNA secondary structure alignments: a progressive profile approach. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 1(1):53–62.

[51] Holder, L., Cook, D., and Djoko, S. (1994). Substructure discovery in the subdue system. In *Proc. of the AAAI Workshop on Knowledge Discovery in Databases*, pages 169–180.

[52] Horvath, S. and Dong, J. (2008). Geometric interpretation of gene coexpression network analysis. *PLoS Computational Biology*, 4(8).

[53] Hu, H., Yan, X., Huang, Y., Han, J., and Zhou, X. (2005). Mining coherent dense subgraphs across massive biological networks for functional discovery. *Bioinformatics*, 21(1):213–221.

[54] Huan, J., Wang, W., Prins, J., and Yang, J. (2004). Spin: mining maximal frequent subgraphs from graph databases. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 581–586. ACM New York, NY, USA.

[55] Huang, Y., Li, H., Hu, H., Yan, X., Waterman, M., Huang, H., and Zhou, X. (2007). Systematic discovery of functional modules and context-specific functional annotation of human genome. *Bioinformatics*, 23(13):i222.

[56] Jiang, T., Lawler, E., and Wang, L. (1994). Aligning sequences via an evolutionary tree: complexity and approximation. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 760–769. ACM New York, NY, USA.

[57] Jiang, T., Lin, G., Ma, B., and Zhang, K. (2002). A general edit distance between RNA structures. *Journal of Computational Biology*, 9(2):371–388.

[58] Jiang, T., Wang, L., and Zhang, K. (1995). Alignment of trees: an alternative to tree edit. *Theoretical Computer Science*, 143(1):137–148.

[59] Jin, R., Wang, C., Polshakov, D., Parthasarathy, S., and Agrawal, G. (2005). Discovering frequent topological structures from graph datasets.

In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 606–611. ACM New York, NY, USA.

[60] Karypis, G. and Kumar, V. (1999). A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359.

[61] Kashtan, N., Itzkovitz, S., Milo, R., and Alon, U. (2004). Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics*, 20(11):1746–1758.

[62] Kawano, S., Hashimoto, K., Miyama, T., Goto, S., and Kanehisa, M. (2005). Prediction of glycan structures from gene expression data based on glycosyltransferase reactions. *Bioinformatics*, 21(21):3976–3982.

[63] Keselman, D. and Amir, A. (1994). Maximum agreement subtree in a set of evolutionary trees-metrics and efficient algorithms. In *Annual Symposium on Foundations of Computer Science*, volume 35, pages 758–758. IEEE Computer Society Press.

[64] Khanna, S., Motwani, R., and Yao, F. (1995). Approximation algorithms for the largest common subtree problem.

[65] Kohonen, T. (1995). Self-organizing maps. *Springer, Berlin*.

[66] Koyuturk, M., Grama, A., and Szpankowski, W. (2004a). An efficient algorithm for detecting frequent subgraphs in biological networks. *Bioinformatics*, 20(90001).

[67] Koyuturk, M., Szpankowski, W., and Grama, A. (2004b). Biclustering gene-feature matrices for statistically significant dense patterns. In *2004 IEEE Computational Systems Bioinformatics Conference, 2004. CSB 2004. Proceedings*, pages 480–484.

[68] Kuboyama, T., Hirata, K., Aoki-Kinoshita, K., Kashima, H., and Yasuda, H. (2006). A gram distribution kernel applied to glycan classification and motif extraction. *Genome Informatics Series*, 17(2):25.

[69] Le, S., Owens, J., Nussinov, R., Chen, J., Shapiro, B., and Maizel, J. (1989). RNA secondary structures: comparison and determination of frequently recurring substructures by consensus. *Bioinformatics*, 5(3):205–210.

[70] Lee, H., Hsu, A., Sajdak, J., Qin, J., and Pavlidis, P. (2004). Coexpression analysis of human genes across many microarray data sets. *Genome Research*, 14(6):1085–1094.

[71] Lemmens, K., Dhollander, T., De Bie, T., Monsieurs, P., Engelen, K., Smets, B., Winderickx, J., De Moor, B., and Marchal, K. (2006). Inferring transcriptional modules from ChIP-chip, motif and microarray data. *Genome biology*, 7(5):R37.

[72] Li, H., Marsolo, K., Parthasarathy, S., and Polshakov, D. (2004). A new approach to protein structure mining and alignment. *Proceedings of the ACM SIGKDD Workshop on Data Mining and Bioinformatics (BIOKDD)*, pages 1–10.

[73] Li, X., Foo, C., and Ng, S. (2007). Discovering protein complexes in dense reliable neighborhoods of protein interaction networks. In *Computational Systems Bioinformatics: Proceedings of the CSB 2007 Conference*, page 157. Imperial College Press.

[74] Liu, N. and Wang, T. (2006). A method for rapid similarity analysis of RNA secondary structures. *BMC bioinformatics*, 7(1):493.

[75] Loß, A., Bunsmann, P., Bohne, A., Loß, A., Schwarzer, E., Lang, E., and Von der Lieth, C. (2002). SWEET-DB: an attempt to create annotated data collections for carbohydrates. *Nucleic acids research*, 30(1):405–408.

[76] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297.

[77] Margush, T. and McMorris, F. (1981). Consensusn-trees. *Bulletin of Mathematical Biology*, 43(2):239–244.

[78] Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002). Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827.

[79] Mitchell, J., Cheng, J., and Collins, K. (1999). A box H/ACA small nucleolar RNA-like domain at the human telomerase RNA 3'end. *Molecular and cellular biology*, 19(1):567–576.

[80] Newman, M. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, 69:026113.

[81] Ohtsubo, K. and Marth, J. (2006). Glycosylation in cellular mechanisms of health and disease. *Cell*, 126(5):855–867.

[82] Onoa, B. and Tinoco, I. (2004). RNA folding and unfolding. *Current Opinion in Structural Biology*, 14(3):374–379.

[83] Packer, N., von der Lieth, C., Aoki-Kinoshita, K., Lebrilla, C., Paulson, J., Raman, R., Rudd, P., Sasisekharan, R., Taniguchi, N., and York, W. (2008). Frontiers in glycomics: Bioinformatics and biomarkers in disease. *Proteomics*, 8(1).

[84] Pizzuti, C. and Rombo, S. (2008). Multi-functional protein clustering in ppi networks. In *Bioinformatics Research and Development*, pages 318–330.

[85] Ragan, M. (1992). Phylogenetic inference based on matrix representation of trees. *Molecular Phylogenetics and Evolution*, 1(1):53.

[86] Ravasz, E., Somera, A., Mongru, D., Oltvai, Z., and Barabasi, A. (2002). Hierarchical organization of modularity in metabolic networks.

[87] Sahoo, S., Thomas, C., Sheth, A., Henson, C., and York, W. (2005). GLYDE-an expressive XML standard for the representation of glycan structure. *Carbohydrate research*, 340(18):2802–2807.

[88] Sanderson, M., Purvis, A., and Henze, C. (1998). Phylogenetic supertrees: assembling the trees of life. *Trends in Ecology & Evolution*, 13(3):105–109.

[89] Satuluri, V. and Parthasarathy, S. (2009). Scalable Graph Clustering using Stochastic Flows: Applications to Community Discovery. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 737–746.

[90] Schølkopf, B. and Smola, A. (2002). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press.

[91] Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., and Friedman, N. (2003). Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature genetics*, 34(2):166–176.

[92] Selkow, S. (1977). The tree-to-tree editing problem. *Information processing letters*, 6(6):184–186.

[93] Shapiro, B. and Zhang, K. (1990). Comparing multiple RNA secondary structures using tree comparisons. *Bioinformatics*, 6(4):309–318.

[94] Sharan, R. and Shamir, R. (2000). CLICK: A clustering algorithm with applications to gene expression analysis. 8:307–316.

[95] Shasha, D., Wang, J., and Zhang, S. (2004). Unordered tree mining with applications to phylogeny. In *in Proceedings of International Conference on Data Engineering*, pages 708–719.

[96] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905.

[97] Shih, F. and Mitchell, O. (1989). Threshold decomposition of gray-scale morphology into binarymorphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(1):31–42.

[98] Smith, T. and Waterman, M. (1981). Identification of common molecular subsequences. *J. Mol. Bwl*, 147:195–197.

[99] Sneath, S. (1973). Hierarchical clustering.

[100] Stark, C., Breitkreutz, B., Reguly, T., Boucher, L., Breitkreutz, A., and Tyers, M. (2006). BioGRID: a general repository for interaction datasets. *Nucleic acids research*, 34(Database Issue):D535.

[101] Stockham, C., Wang, L., and Warnow, T. (2002). Statistically based postprocessing of phylogenetic analysis by clustering. *Bioinformatics*, 18(3):465–469.

[102] Stuart, J., Segal, E., Koller, D., and Kim, S. (2003a). A gene-coexpression network for global discovery of conserved genetic modules. *Science*, 302(5643):249–255.

[103] Stuart, J., Segal, E., Koller, D., and Kim, S. (2003b). A gene-coexpression network for global discovery of conserved genetic modules. *Science*, 302(5643):249–255.

[104] Tai, K. (1979). The tree-to-tree correction problem. *Journal of the Association for Computing Machm© ry*, 26(3):422–433.

[105] Tanay, A., Sharan, R., Kupiec, M., and Shamir, R. (2004). Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proceedings of the National Academy of Sciences*, 101(9):2981–2986.

[106] Tanay, A., Sharan, R., and Shamir, R. (2002). Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18(Suppl 1):S136–S144.

[107] Tinoco, I. and Bustamante, C. (1999). How RNA folds. *Journal of molecular biology*, 293(2):271–281.

[108] Ueda, N., Aoki, K., and Mamitsuka, H. (2004). A general probabilistic framework for mining labeled ordered trees. In *Proceedings of the Fourth SIAM International Conference on Data Mining*, pages 357–368.

[109] Ueda, N., Aoki-Kinoshita, K., Yamaguchi, A., Akutsu, T., and Mamitsuka, H. (2005). A probabilistic model for mining labeled ordered trees: Capturing patterns in carbohydrate sugar chains. *IEEE Transactions on Knowledge and Data Engineering*, 17(8):1051–1064.

[110] Valiente, G. (2002). *Algorithms on trees and graphs*. Springer.

[111] Wang, C. and Parthasarathy, S. (2004). Parallel algorithms for mining frequent structural motifs in scientific data. In *Proceedings of the 18th annual international conference on Supercomputing*, pages 31–40. ACM New York, NY, USA.

[112] Wang, L., Jiang, T., and Gusfield, D. (1997). A more efficient approximation scheme for tree alignment. In *Proceedings of the first annual international conference on Computational molecular biology*, pages 310–319. ACM New York, NY, USA.

[113] Wang, L., Jiang, T., and Lawler, E. (1996). Approximation algorithms for tree alignment with a given phylogeny. *Algorithmica*, 16(3):302–315.

[114] Yamanishi, Y., Bach, F., and Vert, J. (2007). Glycan classification with tree kernels. *Bioinformatics*, 23(10):1211.

[115] Yan, X., Mehan, M., Huang, Y., Waterman, M., Yu, P., and Zhou, X. (2007). A graph-based approach to systematically reconstruct human transcriptional regulatory modules. *Bioinformatics*, 23(13):i577.

[116] You, C. H., Holder, L. B., and Cook, D. J. (2006). Application of graph-based data mining to metabolic pathways. *Data Mining Workshops, International Conference on*, 0:169–173.

[117] Zaki, M. (2005). Efficiently mining frequent trees in a forest: Algorithms and applications. *IEEE Transactions on Knowledge and Data Engineering*, 17(8):1021–1035.

[118] Zhang, K. and Jiang, T. (1994). Some MAX SNP-hard results concerning unordered labeled trees. *Information Processing Letters*, 49(5):249–254.

[119] Zhang, K. and Shasha, D. (1989). Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, 18:1245.

[120] Zhang, S. and Wang, T. (2008). Discovering Frequent Agreement Subtrees from Phylogenetic Data. *IEEE Transactions on Knowledge and Data Engineering*, 20(1):68–82.