Post-Processing Summarization for Mining Frequent Dense Subnetworks

San Ha Seo
Department of Computer Science
Fargo, North Dakota, USA
sanha.seo@ndsu.edu

Saeed Salem
Department of Computer Science
Fargo, North Dakota, USA
saeed.salem@ndsu.edu

ABSTRACT

Gene expression data for multiple biological and environmental conditions is being collected for multiple species. Functional modules and subnetwork biomarkers discovery have traditionally been based on analyzing a single gene expression dataset. Research has focused on discovering modules from multiple gene expression datasets. Gene coexpression network mining methods have been proposed for mining frequent functional modules. Moreover, biclustering algorithms have been proposed to allow for missing coexpression links. Existing approaches report a large number of edgesets that have high overlap. In this work, we propose an algorithm to mine frequent dense modules from multiple coexpression networks using a post-processing data summarization method. Our algorithm mines a succinct set of representative subgraphs that have little overlap which reduce the downstream analysis of the reported modules. Experiments on human gene expression data show that the reported modules are biologically significant as evident by Gene Ontology molecular functions and KEGG pathways enrichment.

KEYWORDS

coexpression networks, functional modules, subnetwork biomarkers, gene expression

ACM Reference Format:
San Ha Seo and Saeed Salem. 2020. Post-Processing Summarization for Mining Frequent Dense Subnetworks. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB '20), September 21–24, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3388440.3415989

1 INTRODUCTION

Breakthroughs in RNA-sequencing and high-throughput technologies have made it possible to collect and analyze massive amount of gene expression data. Gene expression data analysis is an effective way to understand gene function and gene regulation. Conventional clustering methods such as k-means, hierarchical, and biclustering approaches have been used with limited success [11]. Clustering approaches designed specifically for gene expression data have been proposed and were shown to be more effective than conventional methods on some datasets [10, 19]. Recent research focus on the analysis of gene coexpression networks. One of the common approaches for gene expression data anaylsis is to cluster genes based on coexpression, as coexpressed genes tend to be co-functional and co-regulated and clustering genes based on coexpression has proven useful in gene function prediction and regulatory motif identification [2, 4].

Gene expression data often contain a lot of noise due to the complex procedure of microarray experiments, resulting in a high number of spurious coexpression links [7]. Additionally, the simultaneous perturbation of multiple biological pathways in the particular experiment may cause coexpressions that have no biological relevance [8]. The spurious coexpression links often cause the discovery of false gene modules. To overcome this problem, recent studies have focused on integrating multiple gene expression datasets. The goal of these studies is to mine gene clusters that appear in multiple datasets, based on the expectation that biological modules are active across multiple datasets. Graph-theoretic approaches are often used in these studies. Each gene expression dataset is represented as a gene coexpression network, which is a graph where the nodes correspond to genes and the edges correspond to coexpression links between the genes. One approach to extract gene modules from multiple gene expression networks is to mine frequently occurring subnetworks in the set of multiple coexpression networks.

Gene coexpression networks have a property that each node has a unique label. This property can be utilized to design algorithms to avoid the subgraph isomorphism problem, which introduces challenges for the general subgraph mining methods. A number of pattern enumeration algorithms to mine frequent modules from a set of graphs have been proposed [9, 12, 20]. The pattern enumeration algorithms, however, do not scale well when applied to large biological networks, especially when the size of the frequent modules are themselves large. Moreover, the edges must appear in the same supporting networks and missing edges are not allowed, which introduce additional challenges in mining frequent subnetworks. To address these issues, several studies have focused on combining the networks into a summary graph and mining modules in the summary graph. Lee et al. [13] proposed a method to build a summary graph by combining coexpression links that appear frequently accross multiple coexpression networks, and applied hierarchical clustering and the MCODE [1] algorithm to mine highly connected modules in the summary graph. It was shown that the coexpression links that appear in multiple datasets are more likely to represent known functional modules. However, clustering the

BCB '20, September 21–24, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7964-9/20/09…$15.00
https://doi.org/10.1145/3388440.3415989

summary graph directly may result in mining false positive modules. The edges in the false positive modules may be scattered across the graphs such that these modules are neither frequent nor dense in any of the original graphs, and yet they are frequent and dense in the summary graph [7].

Several algorithms have been proposed to overcome these limitations. The CODENSE [7] algorithm is a two-step approach that mines coherent dense subgraphs across a set of multiple graphs. It mines dense subgraphs from the summary graph, similar to the approach in [13]. A second order graph is generated for each dense graph where edges in the second order graph denote high occurrence similarity. In the second step, dense subgraphs are mined from the second-order graphs. Due to the property that a coherent subgraph's second-order graph must be dense, the CODENSE algorithm is not affected by the false positive module problem.

Huang et al. [8] proposed an algorithm to mine frequent subgraphs in a set of multiple graphs using frequent itemset mining approach. Frequent edgesets are mined using an approach similar to itemset mining and then each frequent edgeset serve as a seed for a simulated annealing based biclustering algorithm which maximizes an objective function such that the extracted biclusters are large and have high density of ones. Finally, the connected components in the subgraphs induced by the biclusters are returned as the final output. These modules are frequent but may not be dense.

The MFMS [16] algorithm mines maximal frequent collections of cliques and percolated cliques from a set of multiple graphs. A hybrid graph approach is used in [15, 17], where a weighted graph is constructed. In the hybrid graph, nodes correspond to the original edges in the coexpression networks, and a combined score based on the topological similarity between the edges and the occurrence similarity is used to determine the weight between two edges. Dense subgraphs are then extracted from the weighted graph.

In [18], we have proposed a two-step algorithm that mines approximate frequent dense subgraphs across a set of multiple gene coexpression networks. An approximate frequent dense subgraph is a frequent dense subgraph that may contain noise. The first step is to construct a binary edge occurrence matrix from a set of gene coexpression networks and mine biclusters with high density of ones in the matrix. Each bicluster corresponds to an approximate frequent edgeset. The second step is to mine dense modules from the subgraphs induced by the approximate frequent edgesets. The biclustering step in the algorithm returns a huge number of frequent edgesets, especially for lower support threshold values. The large number of edgesets poses a challenge in analysis. Furthermore, the algorithm produces a large number of duplicate modules in the final set of frequent dense modules due to the edgesets having high overlap with each other.

In this work, we propose an algorithm to address this problem. An overview of the steps of our proposed approach is shown in Figure 1. Our algorithm mines approximate frequent dense subgraphs from a set of multiple gene coexpression networks using a post-processing data summarization to reduce the number of reported edgesets. In the first step, we mine approximate frequent edgesets using a biclustering algorithm; This is similar to the same the first step in [18]. To reduce the number of frequent edgesets, we mine a set of representative frequent edgesets from the set of all frequent edgesets with a post-processing data summarization approach, which uses the concept of dominating set. In the second step, we mine dense modules from the subgraphs induced by the representative frequent edgesets. By mining representative frequent edgesets, we significantly reduce the number of reported edgesets and modules while not losing much information. We conducted experiments on human gene expression data and the extracted modules are shown to be biologically significant.

2 PROBLEM DESCRIPTION

We model gene coexpression networks as undirected, unweighted graphs. Since each gene occurs at most once in a gene coexpression network, a coexpression network is modelled as a relation graph, where each node has a unique label. A relation graph set is a set of graphs that share a common set of nodes.

Relation Graph Set: A relation graph set is a set of n graphs $\mathcal{G} = \{G_1, G_2, \ldots, G_n\}$ where $G_i = (V, E_i)$, V is the set of vertices, E_i is the set of edges for G_i, and $E_i \subseteq V \times V$. All the graphs in the set share the same set of vertices V.

Summary Graph and Edge Occurrence Matrix: Given a relation graph set $\mathcal{G} = \{G_1, G_2, \ldots, G_n\}$ where $G_i = (V, E_i)$. The union of all the edges in the graphs is denoted as E, $E = \{e_1, e_2, \ldots, e_m\} = \bigcup_{i=1}^{n} E_i$. The edge occurrence matrix $\mathcal{B}$ is an $m \times n$ binary matrix where $\mathcal{B}_{ij} = 1$ if $e_i \in E_j$; 0 otherwise. The relation graph set is represented as $\mathcal{G} = (V, E, \mathcal{B})$. The graph set is represented as a summary graph (V, E) and an associated edge attribute matrix $\mathcal{B}$, whose rows correspond to the edges' attributes.

Edge-Induced Subgraph: Given a graph $G(V, E)$ and an edgeset $E' \subseteq E$, the edge-induced subgraph $G'(V', E')$ of G (induced by edgeset E' and denoted as $G[E']$) is a graph whose edgeset is E' and the node set is all the nodes that are the endpoints of the edges.

Note that an edge-induced subgraph does not have isolated nodes since each node that is present in the induced subgraph is an endpoint of at least one edge. Since an edge-induced subgraph is uniquely identified by its edgeset, we refer to the frequent edge-induced subgraph as a frequent edgeset.

Frequent Subgraph: Given a graph set $\mathcal{G} = \{G_1, G_2, \ldots, G_n\}$, a minimum support threshold *minsup*, an edge-induced subgraph G' is a frequent subgraph if it is a subgraph of at least *minsup* graphs. A subgraph $G'(V', E')$ is a subgraph of $G = (V, E)$, denoted as $G' \subseteq G$, if $V' \subseteq V$ and $E' \subseteq E$. The supporting graphs of a subgraph is the set of graphs in which the subgraph appears.

$$sup(G', \mathcal{G}) = \{G_{i1}, G_{i2}, \cdots, G_{ik}\}$$

such that G' is a subgraph of G_i for each G_i in $sup(G', \mathcal{G})$ and k is the number of graphs in which the subgraph appears. When the graph dataset is understood from the context, we denote $sup(G', \mathcal{G})$ simply as $sup(G')$. A subgraph G' is frequent in a graph set $\mathcal{G}$ if

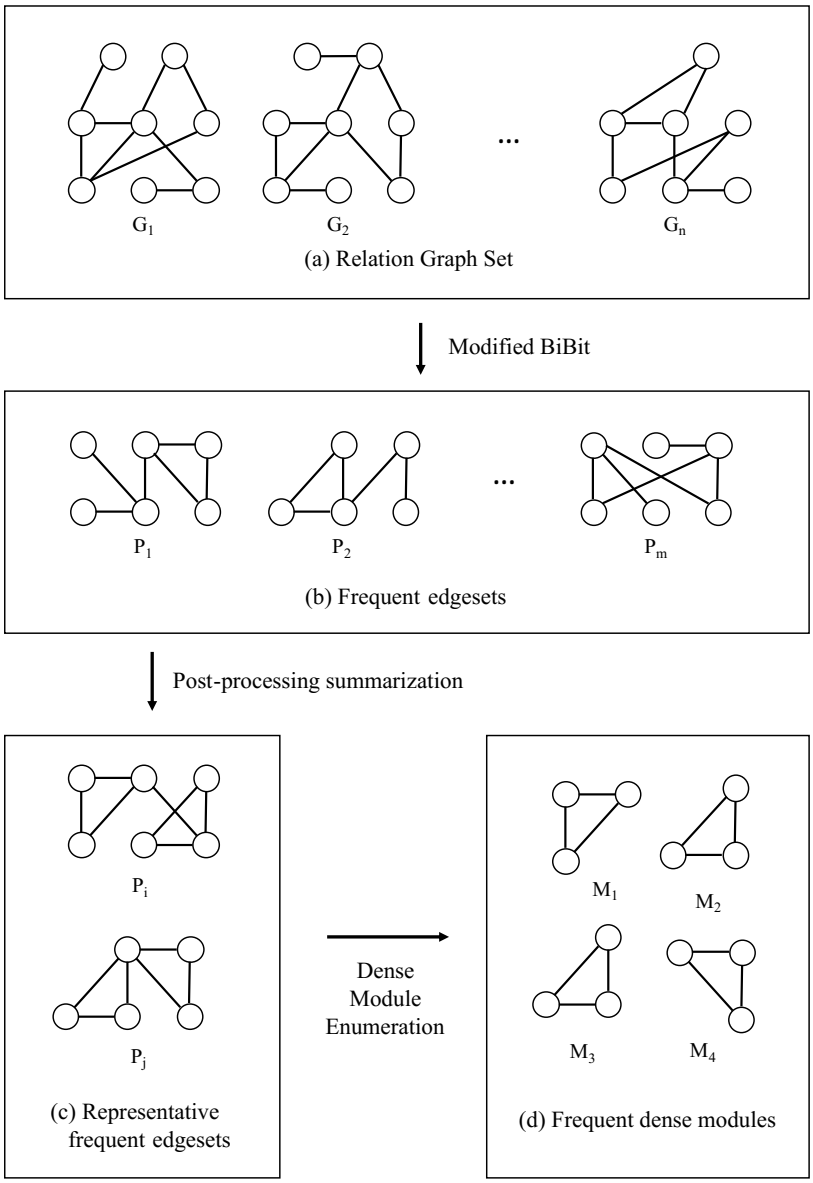


Figure 1: Steps to mine frequent dense modules from a relation graph set: (a) A relation graph set of n graphs; (b) Set of all frequent edgesets/subgraphs mined using the modified BiBit algorithm; (c) Set of representative frequent edgesets/subgraphs mined using post-processing summarization (dominating set); (d) Final set of frequent dense modules mined using DME algorithm

the number of supporting graphs is at least *minsup* graphs, i.e., $|sup(G', \mathcal{G})| >= minsup$.

The definition of frequent subgraphs requires all the edges of a subgraph to appear in all the supporting graphs. Given that some of edges might be missing from a coexpression network due to noise and correlation cutoff, we should change the definition of the occurrence of a subgraph in a coexpression network. Thus we relax the occurrence constraint and introduce the approximate frequent subgraph that is a relaxed form of the frequent subgraph by allowing missing edges (noise). An edge is approximately supported by a graph set if the edge appears in most of the graphs, and a subgraph is approximately supported by a graph set if all the edges of the subgraph are approximately supported by the same graph set.

Approximate Frequent Subgraph: Given a relation graph set $\mathcal{G} = \{G_1, G_2, \ldots, G_n\}$, a minimum support threshold *minsup*, and a noise ratio r, an edge-induced subgraph $G'[E']$ is an approximate frequent subgraph if and only if there exists a graph set $D \subseteq \mathcal{G}$

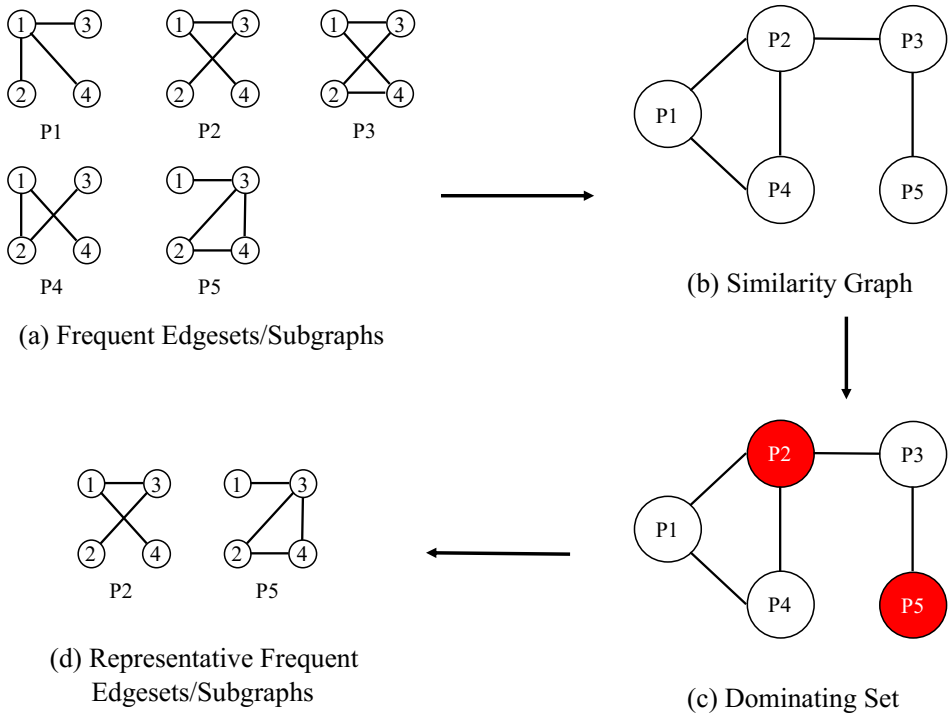


Figure 2: Steps in mining representative frequent subgraphs from set of all frequent subgraphs: (a) Set of all frequent edgesets/subgraphs; (b) Similarity graph for the set in (a) with similarity threshold 0.5 (Jaccard similarity coefficient); (c) Dominating set of the similarity graph in (b); (d) Set of representative frequent edgesets/subgraphs

such that $|D| >= minsup$ and for every edge $e \in E'$, e occurs in at least $\lfloor |D| * (1-r) \rceil$ graphs in D, the nearest integer to $|D| * (1-r)$.

The noise ratio r is a real number between 0 and 1, controlling the ratio of the supporting graphs an edge can be missed. An edge e need not be present in every graph in D. For the remainder of this paper, we refer to the approximate frequent edgesets/subgraphs simply as frequent edgesets/subgraphs.

The set of all frequent subgraphs is large considering the combinatorial nature of the frequent subgraphs. Moreover, these subgraphs have high overlap since two frequent subgraphs can differ by only one or two edges. Therefore, we mine a representative set of these approximate frequent subgraphs.

In the proposed algorithm, we mine a set of representative edgesets. A set of representative edgesets is a subset of the set of edgesets such that every edgeset not included in the representative set has at least one similar edgeset in the representative set. Moreover, we are interested in dense subgraphs in these representative frequent subgraphs as these edge-induced subgraphs are not necessarily dense.

Set of Representative Edgesets: Given a set of edgeset patterns P and an edgeset similarity threshold s, a subset $P' \subseteq P$ is a set of representative edgesets if for every edgeset $E \in P \setminus P'$, there exists an edgeset $E' \in P'$ such that $sim(E, E') \geq s$, where $sim(E, E')$ is the similarity between the two sets. Each edgeset in P is either in P' or is similar to an edgeset in P'.

Graph Density: The density of a graph G is $2m/(n(n-1))$ where m is the number of edges and n is the number of nodes in G. Given a density threshold α, a subgraph G is dense if its density is greater than or equal to α.

Dense Frequent Subgraphs: Each representative frequent edgeset induces a graph from the summary graph. This subgraph is approximately frequent. The dense subgraphs in the induced subgraph are the reported modules. These modules are also frequent since they are extracted from the frequent subgraph whose edges are approximately frequent.

3 MINING FREQUENT DENSE MODULES

In this work, we mine frequent dense subgraphs from a set of gene coexpression networks. Given a relation graph set $\mathcal{G}$, minimum support threshold, noise threshold, edgeset similarity threshold, and density threshold, our task is to find subgraphs that are both approximate frequent and dense. One approach for mining representative pattern is online data summarization where the data is processed as they are produced. In this case, the edgesets are processed as they are mined by the biclustering algorithm. Beginning with an empty set of representative edgesets, when the biclustering algorithm encounters a frequent edgeset, the edgeset is added

to the representative set if there is no similar edgeset (based on a user-defined similarity threshold) already in the set. As a result, the final set contains edgesets such that every frequent edgeset not in the set has at least one similar edgeset in the set. While the online data summarization approach is efficient, it does not return the optimal result since it depends on the order in which the edgesets are discovered. We propose a two-step approach to mine approximate frequent dense subgraphs. In the first step, we mine frequent edgesets using a biclustering approach. In order to reduce the number of reported patterns and decrease the overlap between the reported patterns, we use a post-processing data summarization method to mine a set of representative frequent edgesets. In the proposed post-processing data summarization method, all frequent edgesets are first mined and then a subset of these frequent edgesets is chosen such that every edgeset not in the set has at least one similar edgeset in the set. The summarization method uses the concept of similarity graph and dominating set to choose the representative frequent edgesets [6].

3.1 Mining Representative Frequent Edgesets

We employ the modified BiBit algorithm described in [18] to mine all frequent edgesets from a relation graph set. The modified BiBit algorithm is a biclustering algorithm which mines biclusters with high density of ones from a binary matrix. Each bicluster corresponds to an approximate frequent edgeset that allows missing edges. The modified Bibit procedure is called in line 1 in Algorithm 1. The procedure takes an $m \times n$ binary edge occurrence matrix $\mathcal{B}$ in which the rows correspond to edges and the columns correspond to graphs. The minimum number of rows, *minSize*, correpsonds to the minimum edgeset size, and the minimum number of columns, *minsup*, corresponds to the minimum number of supporting graphs threshold. The result is a bicluster with density of ones greater than or equal to $1 - r$. The row set of the bicluster represents a frequent edgeset.

3.2 Similarity Measure

We use the Jaccard similarity coefficient to measure the similarity between edgesets. The Jaccard similarity coefficient between two sets is defined as the cardinality of the intersection of the two sets divided by the cardinality of the union of the two sets. More precisely, the Jaccard similarity coefficient of the two sets A and B is

$$sim(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

The similarity score ranges between 0 and 1. Roughly, it is the measure of the degree of overlap between the two sets, with 0 indicating no similarity and 1 indicating identical sets. In general, the size of the representative set is smaller for lower value of edgeset similarity threshold. For the special case when the similarity threshold is set to 1, the set of representative frequent edgesets is the same as the set of all frequent edgesets. And for the special case when the similarity threshold is set to 0, the first encountered frequent edgeset is the only pattern in the set, as it is 'similar' to all other edgesets.

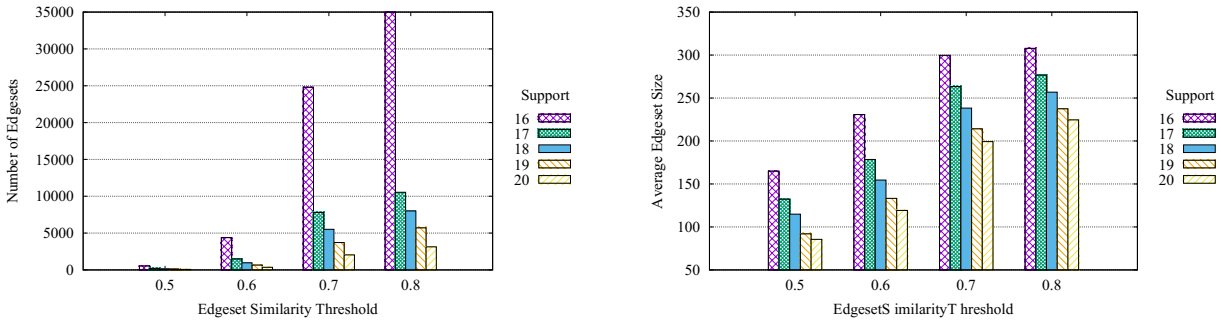
Algorithm 1: Mining Representative Frequent Dense Modules

Input : $\mathcal{G} = (V, E, \mathcal{B})$: A relation graph set of n graphs
minSize: minimum number of rows
minsup: minimum number of columns
r: noise ratio
s: edgeset similarity threshold
α: module density threshold
Output : $\mathcal{X}$: Dense Frequent Modules

```
   /* Mining Approximate Frequent Edgesets                    */
1  P = runModifiedBibit(B, minSize, minsup, r)
   /* Constructing Similarity Graph                           */
2  V_P ← {1, 2, ··· , |P|}
3  E_P ← ∅
4  for i ← 1 to |P| do
5      for j ← i + 1 to |P| do
6          if sim(P_i, P_j) ≥ s then
7              E_P = E_P ∪ {(i, j)}
8  G = (V_P, E_P)
   /* Extracting Dominating Set of G                          */
   /* Initially each node dominates all neighbors             */
9  S ← ∅
10 du(v) = |Neighbors(v)|, for each v ∈ V_P
11 while there exists undominated nodes do
12     v ← the vertex that dominates the most nodes
13     S = S ∪ {v}
14     mark all neighbors of v as dominated
15     update the number of undominated nodes that each node
         dominates
   /* Extracting Dense Modules                                */
16 X ← ∅
17 for i ← 1 to |S| do
18     g_i = G[S_i]                          // edge Induce Subgraph
19     X = X ∪ DME(g_i, α)
20 return X
```

3.3 Similarity Graph

Once we mine the set of all frequent edgesets, we construct the similarity graph to represent the similarities between the edgesets. In the similarity graph, each node corresponds to an edgeset, and there is an edge between two nodes if the similarity between the two corresponding edgesets exceeds a user-defined similarity threshold. More formally, given a set of m frequent edgeset patterns $P = \{P_1, P_2, \cdots, P_m\}$ and a user-defined similarity threshold s, the similarity graph $G_P(V_P, E_P)$ is a graph such that each node $v_i \in V_P$ corresponds to pattern $P_i \in P$ and there is an edge $(v_i, v_j) \in E_P$ if $sim(P_i, P_j) \geq s$, where $sim(P_i, P_j)$ is the similarity between patterns P_i and P_j. Figure 2 (b) shows the similarity graph constructed from the set of edgeset patterns in (a) with similarity threshold 0.5. For example, the similarity graph in (b) has the edge (P_1, P_2) because the similarity between edgesets P_1 and P_2 is 0.5. Constructing the similarity graph for the frequent edgesets is described in lines 2-8 in Algorithm 1.



(a) Number ofEdgesets vs. Similarity Threshold

(b) AverageEdgeset Size vs. Similarity Threshold

Figure 3: Number of frequent edgesets and average edgeset size for varying edgeset similarity threshold values

3.4 Dominating Set

A dominating set of a graph $G(V, E)$ is a subset $S \subseteq V$ such that every node not in S is connected to at least one node in S. A minimum dominating set is the smallest such set. A graph can have multiple minimum dominating sets. Since the similarity graph for the set of all frequent edgesets represents edgeset similarities, a minimum dominating set of the similarity graph is the smallest node set which corresponds to the set of representative frequent edgesets. Figure 2 (c) shows a minimum dominating set for the the similarity graph in (b). The corresponding representative frequent edgesets are shown in (d). For the similarity graph $G_P(V_P, E_P)$, the goal is to find a subset of vertices (patterns) $S_P \subseteq V_P$ that dominates all the remaining vertices (patterns). The problem of finding a minimum dominating set of a graph is NP-hard. There are linear reductions between the set cover problem, a well-known NP-hard problem, and the minimum dominating set problem [3]. Therefore, we employ an approximation greedy algorithm whose solution is optimal up to a certain factor. The greedy algorithm starts with an empty set, $S = \emptyset$, and adds vertices to S until S is a dominating set of the graph. The most common greedy algorithm is to select the vertex that has the maximum number of neighbors that are not dominated. The number of undominated vertices that a vertex v dominates is denoted by $du(v)$. Initially each vertex dominates itself and its neighbors. So the vertex with the largest degree is chosen as the first vertex to add to S. Lines 9-15 in Algorithm 1 shows the greedy approach for finding the dominating set. Next the du score is updated for all vertices and the algorithm selects the vertex with the largest du score. If there are multiple vertices with the largest score, a vertex is chosen randomly. This process is repeated until all vertices are dominated. For the similarity graph in Figure 2 (b), the greedy algorithm selects P_2 as the first vertex in the dominating set S since P_2 dominates four vertices including itself. After updating the du score, both P_3, and P_5 have the same score of 1. The algorithm chooses one of them randomly. Note that P_3 is still a candidate to be added to the dominating set even though it is dominated. The algorithm then selects P_5 and terminates since all vertices are dominated now. The final dominating set is $S = \{P_2, P_5\}$, indicating that the corresponding patterns are the representative frequent edgesets.

3.5 Extracting Dense Modules

The final step is to extract the dense subgraphs for each representative edgeset. We employ the dense module enumeration (DME) algorithm on the edge-induced subgraph for each representative edgeset [5]; the DME algorithm is called in lines 16-19 in Algorithm 1.

4 EXPERIMENTS

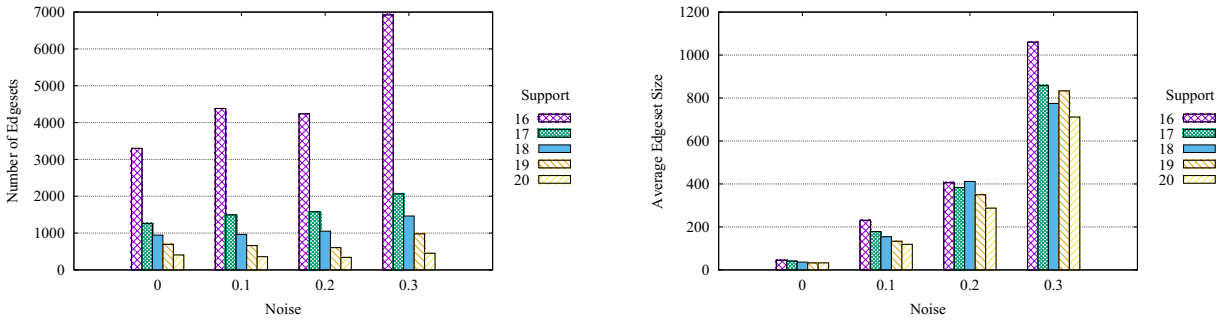
In order to assess the effectiveness of our algorithm, we mined the representative frequent edgesets and frequent dense modules from 35 tissue gene coexpression networks constructed by the Genetic Network Analysis Tool [14]. The gene coexpression networks were constructed from the gene expression of non-diseased tissue samples from Genotype-Tissue Expression (GTEx) data [1]. Each coexpression network contains 9,998 genes and 14,415 links on average. In total, there are 1,548,622 unique links that appear in at least one network and 4,127 edges that appear in at least 20 networks. On average, each link appears in 3.28 networks.

4.1 Effect of Data Summarization

To assess the effectiveness of the proposed approach, we first ran the modified BiBit algorithm on the binary edge occurrence matrix constructed from the 35 gene coexpression networks for support thresholds $minsup = 16, 17, 18, 19, 20$ and noise threshold $r = 0.1$. Then we applied the post-processing method on the discovered frequent edgesets for edgeset similarity thresholds $s = 0.5, 0.6, 0.7, 0.8$. Figure 3 shows how the number and the average size of the representative frequent edgesets change with various edgeset similarity and support threshold values. As shown in the figure, the number of representative frequent edgesets increases as the support threshold decreases and as the edgeset similarity threshold increases. The average size of the representative frequent edgesets increases as the edgeset similarity threshold increases.

To evaluate the effect of the post-processing summarization for frequent edgesets, we mined the frequent frequent edgesets for $minsup = 20$, and used edgeset similarity thresholds 0.5 to 0.8 for mining representative frequent edgesets. The number of reported frequent edgesets for various edgeset similarity thresholds

[1] https://www.gtexportal.org/



(a) Number of Edgesets vs. Noise Ratio

(b) AverageEdgeset Size vs. Noise Ratio

Figure 4: Number of frequent edgesets and average edgeset size for varying noise ratio values

Table 1: Comparison of the number of edgesets for support 20 for varying similarity thresholds

noise	0	0.1	0.2	0.3
Without summarization	3,004	3,153	3,224	3,244
With summarization ($s = 0.5$)	113	73	62	81
With summarization ($s = 0.6$)	407	360	341	453
With summarization ($s = 0.7$)	1,226	2,044	2,310	2,894
With summarization ($s = 0.8$)	2,693	3,131	3,220	3,243

is shown in Table 1. It shows that the number of representative frequent edgesets increases with increasing similarity threshold. For a large similarity threshold, fewer edgesets are similar to each other and therefore the number of representative patterns is larger. For a small similarity threshold, less number of patterns is needed to represent the entire set.

4.2 Topological Analysis of Frequent Edgesets and Frequent Dense Modules

To perform the topological analysis of the representative frequent edgesets, we mined representative frequent edgesets for for support thresholds $minsup = 16, 17, 18, 19, 20$, noise thresholds $r = 0, 0.1, 0.2, 0.3$, and edgeset similarity threshold $s = 0.6$. Figure 4 shows how the number and the average size of representative frequent edgesets change with various noise threshold values. It shows that both the number and the average size of the representative frequent edgesets increase as the noise increases. For a larger noise, more edges can be added to the edgeset without violating the noise constraint.

From the subgraphs induced by the frequent edgesets, we mined dense modules using the DME [5] algorithm with density thresholds 0.5 and 0.6. We only considered modules of size four or larger. Table 2 shows the topological properties of the frequent dense modules for support thresholds $minsup = 17, 18, 19, 20$, noise thresholds $r = 0, 0.1, 0.2$, and edgeset similarity threshold $s = 0.6$. The number of edgesets that induces a subgraph that has at least one module is denoted as M'. The average number of dense modules per frequent subgraph is denoted as $\overline{DM}$. Finally, $\overline{V'}$ denotes the average size of the dense modules. The results show that less edgesets contain dense modules and the average number of dense modules is lower for a larger support threshold. At higher support thresholds, edgesets induce sparse graphs that are less likely to contain dense modules.

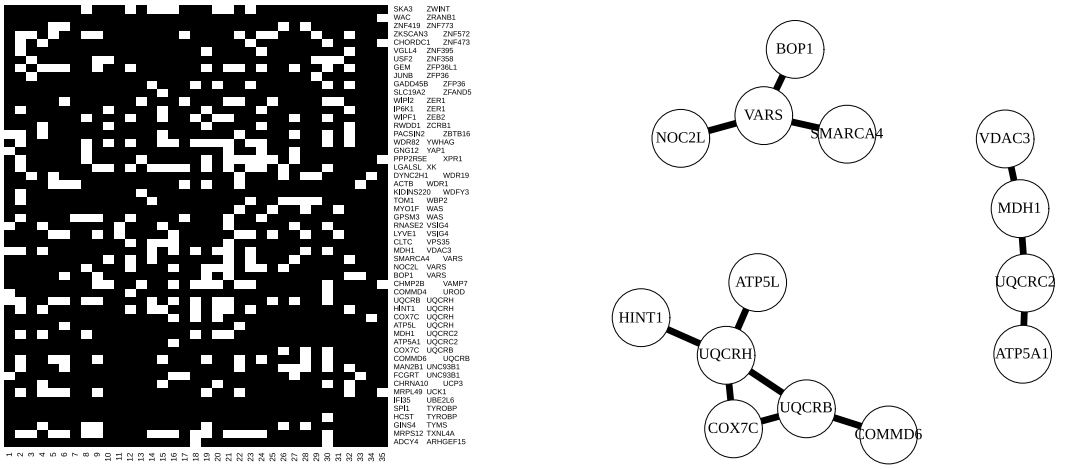
4.3 Gene Ontology Enrichment Analysis

To evaluate the biological significance of the reported modules, we conducted Gene Ontology enrichment analysis on the reported unique frequent dense modules. The results show that the modules are enriched with known KEGG pathways and Gene Ontology molecular functions. A module is enriched if it overlaps with the geneset of a known molecular signature. We used the overrepresentation of genes with a specific annotation in a gene set using the hybergeometric test with $pvalue = 0.01$. For biological terms, we used the KEGG pathway database, which has 186 sets covering 5,241 genes, and the GO Molecular Function Ontology, which has 1,645 sets covering 15,599 genes. Table 3 shows the percentage of frequent dense modules that are biologically enriched. The percentage of enriched modules with GO molecular functions and KEGG pathways is denoted by E_M and E_K, respectively. It shows that frequent dense modules with smaller noise ratio have higher percentage of enrichment. The GO molecular functions have higher enrichment than KEGG pathways, as there are much more molecular functions than KEGG pathways. The frequent dense modules can be enriched with multiple biological annotations, and an annotation can be enriched in multiple frequent dense modules. Table 4 shows the top enriched biological signatures in the reported modules for $sup = 17$, $noise = 0.1$, and $density = 0.5$, and the number of frequent dense modules in which the annotation is enriched.

Figure 5 shows an example of a frequent edgeset for $sup = 19$, $noise = 0.2$. (a) show the submatrix of the binary edge occurrence matrix which represents the edge occurrences in the frequent edgeset in the 35 networks. Each row corresponds to an edge in the edgeset, and each column corresponds to a gene coexpression network. (b) shows the dense modules mined from the subgraph induced by the edgeset for density threshold 0.5. Nodes are labeled by their corresponding gene identifiers. The genes in this representative approximate edgeset are enriched with five KEGG pathways;

Table 2: Topological properties of the frequent dense modules

noise		0			0.1			0.2		
minsup	density	M'	$\overline{DM}$	$\overline{V'}$	M'	$\overline{DM}$	$\overline{V'}$	M'	$\overline{DM}$	$\overline{V'}$
17	0.5	250	10.2	4	1.3 K	43.6	4.2	1.4 K	181.8	4.3
	0.6	9	1	4	312	4.4	4.2	751	15.3	4.4
18	0.5	164	8.8	4	780	35.5	4.1	949	211.7	4.3
	0.6	1	1	4	162	4	4.1	509	17.6	4.4
19	0.5	107	7.1	4	515	28.1	4.1	526	171.1	4.3
	0.6	1	1	4	79	3.3	4.2	262	15.4	4.4
20	0.5	66	7.4	4	269	23.5	4.1	291	130.4	4.2
	0.6	1	1	4	36	2.6	4	128	11.5	4.3



(a) Submatrix for Frequent Edgeset

(b) Frequent Dense Modules

Figure 5: Sample frequent edgeset for $minsup = 19$ **and** $noise = 0.2$**, and dense modules in the edgeset for** $density = 0.5$

Table 3: GO term enrichment analysis for frequent dense modules

noise		0		0.1		0.2	
minsup	density	E_M	E_K	E_M	E_K	E_M	E_K
17	0.5	81.5	61.3	66.8	56.7	51.7	44.2
	0.6	100	33.3	79.8	59.1	71.5	52.7
18	0.5	82.4	62.4	72.7	60.9	52.3	44.2
	0.6	100	0	79.5	56.8	70.2	51.8
19	0.5	86.5	62.5	75.1	63.1	55.6	47
	0.6	100	0	87.5	62.5	74.4	54.2
20	0.5	80.3	57.7	80.3	67.1	62.7	52.2
	0.6	100	0	89.3	75	78.9	56.5

Oxidative Phosphorylation, Cardiac Muscle Contraction, Alzheimers Disease, Parkinsons Disease, and Huntingtons Disease, and two molecular functions; Electron Transfer Activity, and Oxidoreductase Activity.

5 CONCLUSION

Gene Coexpression subnetworks that are present in multiple gene expression datasets improves the functional modules and biomarkers discovery tasks. Mining frequent subnetworks is computationally challenging and results in a large number of overlapping subnetworks. We have proposed a post-processing approach for mining representative frequent dense modules. First, we mine all frequent edgesets. We then construct a similarity graph that captures the similarity between edgesets. We employ a greedy algorithm for extracting the minimum dominating set in the similarity graph. The frequent edgesets corresponding to the nodes in the dominating set are the final representative edgesets. The final step is to mine dense modules from these the induced subgraphs of these frequent edgesets. Because this is a post-processing summarization, it is not dependent on the order in which frequent edgesets are mined. Gene Ontology molecular functions and KEGG pathways enrichment analysis reveals that the frequent dense modules are highly enriched with known biological knowledge. We plan to explore different summarization and clustering technique control the size and quality of the representative set of patterns.

Table 4: Top enriched biological signatures in the reported modules for $minsup = 17$, $noise = 0.1$, and $density = 0.5$

GO Molecular Function	Count
Structural Constituent Of Ribosome	1509
Rrna Binding	389
5s Rrna Binding	219
Electron Transfer Activity	187
Antigen Binding	149
Ubiquitin Protein Transferase Regulator Activity	146
Oxidoreductase Activity Acting On Nad P H	137
Immunoglobulin Receptor Binding	135
Nadh Dehydrogenase Activity	132

KEGG Pathway	Count
Ribosome	1511
Huntingtons Disease	368
Oxidative Phosphorylation	362
Parkinsons Disease	344
Alzheimers Disease	337
Cardiac Muscle Contraction	167
Autoimmune Thyroid Disease	50
Aminoacyl Trna Biosynthesis	46
Mapk Signaling Pathway	43
Leishmania Infection	30

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. RII Track-2 FEC 1826834.

REFERENCES

[1] Gary D. Bader and Christopher W.V. Hogu. 2003. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics* 4, 2 (2003).

[2] Alvis Brazma and Jaak Vilo. 2000. Gene expression data analysis. *FEBS Letters* 480, 1 (2000), 17–24.

[3] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2001. *Introduction to Algorithms* (2nd ed.). The MIT Press.

[4] Michael B. Eisen, Paul T. Spellman, Patrick O. Brown, and David Botstein. 1998. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences* 95, 25 (1998), 14863–14868.

[5] Elisabeth Georgii, Sabine Dietmann, Takeaki Uno, Philipp Pagel, and Koji Tsuda. 2009. Enumeration of condition-dependent dense modules in protein interaction networks. *Bioinformatics* 25, 7 (2009), 933–940.

[6] Aditya Goparaju, Tyler Brazier, and Saeed Salem. 2015. Mining representative maximal dense cohesive subnetworks. *Network Modeling Analysis in Health Informatics and Bioinformatics* 4, 1 (2015), 1–11.

[7] Haiyan Hu, Xifeng Yan, Yu Huang, and Xianghong Jasmine Zhou. 2005. Mining coherent dense subgraphs across massive biological networks for functional discovery. *Bioinformatics* 21 Suppl 1 (2005), i213–i221.

[8] Yu Huang, Haifeng Li, Haiyan Hu, Xifeng Yan, Michael S. Waterman, Haiyan Huang, and Xianghong Jasmine Zhou. 2007. Systematic discovery of functional modules and context-specific functional annotation of human genome. *Bioinformatics* 23, 13 (2007), i222–i229.

[9] Daxin Jiang and Jian Pei. 2009. Mining frequent cross-graph quasi-cliques. *ACM Trans. Knowl. Discov. Data* 2, 4 (jan 2009), 16:1–16:42.

[10] Daxin Jiang, Jian Pei, and Aidong Zhang. 2003. DHC: A Density-Based Hierarchical Clustering Method for Time Series Gene Expression Data. In *Proceedings of the 3rd IEEE Symposium on BioInformatics and BioEngineering (BIBE '03)*. IEEE Computer Society, Washington, DC, USA, 393–.

[11] Daxin Jiang, Chun Tang, and Aidong Zhang. 2004. Cluster Analysis for Gene Expression Data: A Survey. *IEEE Trans. on Knowl. and Data Eng.* 16, 11 (Nov. 2004), 1370–1386.

[12] Mehmet Koyuturk, Ananth Grama, and Wojciech Szpankowski. 2004. An Efficient Algorithm for Detecting Frequent Subgraphs in Biological Networks. *Bioinformatics* 20, Suppl 1 (2004), i200–i207.

[13] Homin K. Lee, Amy K. Hsu, Jon Sajdak, Jie Qin, and Paul Pavlidis. 2004. Coexpression analysis of human genes across many microarray data sets. *Genome Res.* 14, 6 (2004), 1085–1094.

[14] Emma Pierson, the GTEx Consortium, Daphne Koller, Alexis Battle, and Sara Mostafavi. 2015. Sharing and Specificity of Co-expression Networks across 35 Human Tissues. *PLOS Computational Biology* 11, 5 (05 2015), 1–19. https://doi.org/10.1371/journal.pcbi.1004220

[15] Saeed Salem. 2017. Template edge similarity graph clustering for mining multiple gene expression datasets. *International Journal of Data Mining and Bioinformatics* 18, 1 (2017), 28–39.

[16] Saeed Salem and Cagri Ozcaglar. 2013. MFMS: Maximal Frequent Module Set Mining from Multiple Human Gene Expression Data Sets. In *Proceedings of the 12th International Workshop on Data Mining in Bioinformatics* (Chicago, Illinois) *(BioKDD '13)*. ACM, New York, NY, USA, 51–57.

[17] Saeed Salem and Cagri Ozcaglar. 2014. Hybrid coexpression link similarity graph clustering for mining biological modules from multiple gene expression datasets. *BioData Mining* 7, 1 (2014), 16.

[18] San Ha Seo and Saeed Salem. 2020. Mining approximate frequent dense modules from multiple gene expression datasets. In *Proceedings of the 12th International Conference on Bioinformatics and Computational Biology (EPiC Series in Computing, Vol. 70)*, Qin Ding, Oliver Eulenstein, and Hisham Al-Mubaid (Eds.). EasyChair, 129–138.

[19] Roded Sharan and Ron Shamir. 2000. Center CLICK: A Clustering Algorithm with Applications to Gene Expression Analysis. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, 307–316.

[20] Xifeng Yan, Xianghong Jasmine Zhou, and Jiawei Han. 2005. Mining closed relational graphs with connectivity constraints. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining* (Chicago, Illinois, USA) *(KDD '05)*. ACM, New York, NY, USA, 324–333.