Mining representative approximate frequent coexpression 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

Advances in high-throughput microarray and RNA-sequencing technologies have lead to a rapid accumulation of gene expression data for various biological conditions across multiple species. Mining frequent gene modules from a set of multiple gene coexpression networks has applications in functional gene annotation and biomarker discovery. Biclustering algorithms have been proposed to allow for missing coexpression links. Existing approaches report a large number of edgesets which are computationally intensive to analyze, and have high overlap among the reported subnetworks. In this work, we propose an algorithm to mine frequent dense modules from multiple coexpression networks using an online data summarization method. Our algorithm mines a succinct set of representative subgraphs that have little overlap which reduces the downstream analysis of the reported modules. Experiments on human gene expression data show that the reported modules are biologically significant as evident by the high enrichment of GO molecular functions and KEGG pathways in the reported modules.

KEYWORDS

Coexpression networks, Frequent subgraphs, Dense subgraphs

ACM Reference Format:
San Ha Seo and Saeed Salem. 2020. Mining representative approximate frequent coexpression 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, 8 pages. https://doi.org/10.1145/3388440.3415584

1 INTRODUCTION

Advancements in high-throughput microarray and RNA-sequencing technologies enabled the collection and analysis of large amount of gene expression data. Analysis of gene expression data is useful in understanding gene function and gene regulation. Various conventional clustering techniques, including k-means, hierarchical, and biclustering approaches, have been employed with limited success [9]. Several clustering approaches designed specifically for gene expression data have been proposed, and were shown to be more effective than the conventional clustering methods on some datasets [8, 17]. Recent research has focused on analyzing gene coexpression networks. A common approach for analyzing gene expression data is clustering genes based on coexpression. It is believed that coexpressed genes are likely to be co-functional and co-regulated, and clustering genes based on coexpression has proven helpful in predicting unknown gene functions and identifying regulatory motifs [2, 3].

The complex procedure of microarray experiments often causes gene expression data to contain a lot of noise, leading to a significant number of spurious coexpression links [5]. Additionally, some coexpression may be caused by the simultaneous perturbation of multiple biological pathways in the particular experiment rather than by biological relevance [6]. These spurious coexpression links often result in the discovery of false modules (sets of coexpressed genes). To address this problem, recent studies have focused on integrating multiple gene expression datasets for analysis. Based on the expectation that biological modules are active across multiple datasets, these studies aim to discover gene clusters that appear across multiple datasets. Graph-theoretic approaches are commonly used in these studies. Each gene expression dataset is represented as a gene coexpression network, in which nodes correspond to genes and edges correspond to coexpression links between genes. One approach for extracting gene modules from multiple networks is mining frequently occurring coexpression subnetworks in the set of multiple gene coexpression networks.

The gene coexpression networks have unique node labels and this feature has been exploited to design algorithms that avoid the subgraph isomorphism problem that introduces challenges for the general subgraph mining methods. Several pattern enumeration algorithms for mining frequent modules in a set of graphs have been proposed [7, 10, 18]. However, the pattern enumeration algorithms do not scale well when applied to massive biological networks, especially when there are large frequent modules. Moreover, another challenge in mining for frequent subnetworks is that edges have to appear in the same supporting networks and does not allow for missing edges. To overcome the scalability issue, many studies have focused on aggregating the networks into a summary graph and discovering modules in the summary graph. Lee et al. [11] proposed a method that combines frequent coexpression links in multiple coexpression networks to build a summary graph, and applied hierarchical clustering and the MCODE [1] algorithms to mine highly connected modules from the summary graph. The coexpression links that occur across multiple datasets were shown to be more likely to represent known functional modules. Directly clustering the summary graph, however, may lead to the discovery of false positive modules. The edges in these modules may be scattered across the graphs such that they are frequent and dense in

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

ACM ISBN 978-1-4503-7964-9/20/09…$15.00
https://doi.org/10.1145/3388440.3415584

the summary graph, but neither frequent nor dense in any of the original graphs [5].

Numerous algorithms have been proposed to address these limitations. Hu et al. [5] proposed the CODENSE algorithm, a two-step approach that mines coherent dense subgraphs across a set of multiple graphs. The CODENSE algorithm mines dense subgraphs from the summary graph, similar to the approach in [11]. Each dense subgraph is then mapped to a second-order graph, which is a graph whose nodes correspond to the edges of the input graph, and there is an edge between two nodes if there is a high correlation between occurrence of the two corresponding edges across the entire graph set. In the second step, dense subgraphs are mined from the second-order graphs. The CODENSE algorithm overcomes the false positive module problem due to the property that a coherent subgraph's second-order graph must be dense.

Huang et al. [6] proposed an algorithm that mines frequent subgraphs across a set of multiple graphs by using frequent itemset mining approach. The problem is mapped to a frequent itemset mining problem by representing each graph by transactions and each edge by items. Frequent itemset mining technique is used to mine frequent edgesets from the graph set. The frequent edgesets serve as seeds for a biclustering algorithm that uses simulated annealing to maximize an objective function such that the discovered biclusters are large and have high density of ones. The algorithm returns the connected components in the biclusters as the final output. The output modules are frequent but not necessarily dense.

Salem et al. [14] proposed the MFMS algorithm, which mines maximal frequent collections of k-cliques and percolated k-cliques across a set of multiple graphs. The graph set is first mapped to a summary graph with edge attributes. The edge attributes are captured in a binary edge occurrence matrix, where each row corresponds to an edge in the summary graph and each column corresponds to a graph in the graph set, and each entry indicates the presence of the edge in the graph. Maximal frequent edgesets are mined from the graph set using maximal itemset mining approach, and then cliques and percolated cliques are mined from subgraphs induced by the maximal frequent edgesets. In [13, 15], they proposed an approach that constructs a weighted graph whose nodes corresponds to the original edges in the coexpression networks. The weight between two edges is calculated as a combined score based on the topological similarity between the edges and the occurrence similarity.

In Seo et al. [16], we have proposed a two-step algorithm to mine approximate frequent dense subgraphs from a set of multiple coexpression networks. The approximate frequent dense subgraphs are frequent dense subgraphs that may contain some noise. In the first step, a binary edge occurrence matrix is constructed from the set of coexpression networks, and then biclusters with high density of ones are mined from the edge occurrence matrix. Each edgeset bicluster corresponds to a approximate frequent edgeset. In the second step, dense modules are extracted from the subgraphs induced by the frequent edgesets. The first step of the algorithm (biclustering) reports huge number of edgesets, especially for low support thresholds. This makes the analysis very difficult. Moreover, many edgesets have large overlap with each other, producing many duplicate modules in the final set of frequent dense modules.

In this work, we propose an algorithm to mine representative approximate frequent dense subgraphs. The proposed approach integrates the summarization task into the mining process. After the representative frequent edgesets are mined, dense modules are extracted from the subgraphs induced by the representative frequent edgesets. The number of representative frequent edgesets is much less than the number of all frequent edgeset. Experiments on Human gene coexpression networks show that representative frequent dense modules are highly enriched with known biological knowledge.

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 $G_i = (V, E_i)$ and $E_i \subseteq V \times V$. A common set of nodes V is shared by all graphs.

Figure 1 (a) shows an example of a relation graph set of six graphs. Note they share a common set of nodes. We represent the n graphs as a summary graph $G(V, E)$ and an associated binary edge occurrence matrix, $\mathcal{B}$. Each row of the matrix is a binary vector whose entries represent the presence of the edge in the corresponding graphs.

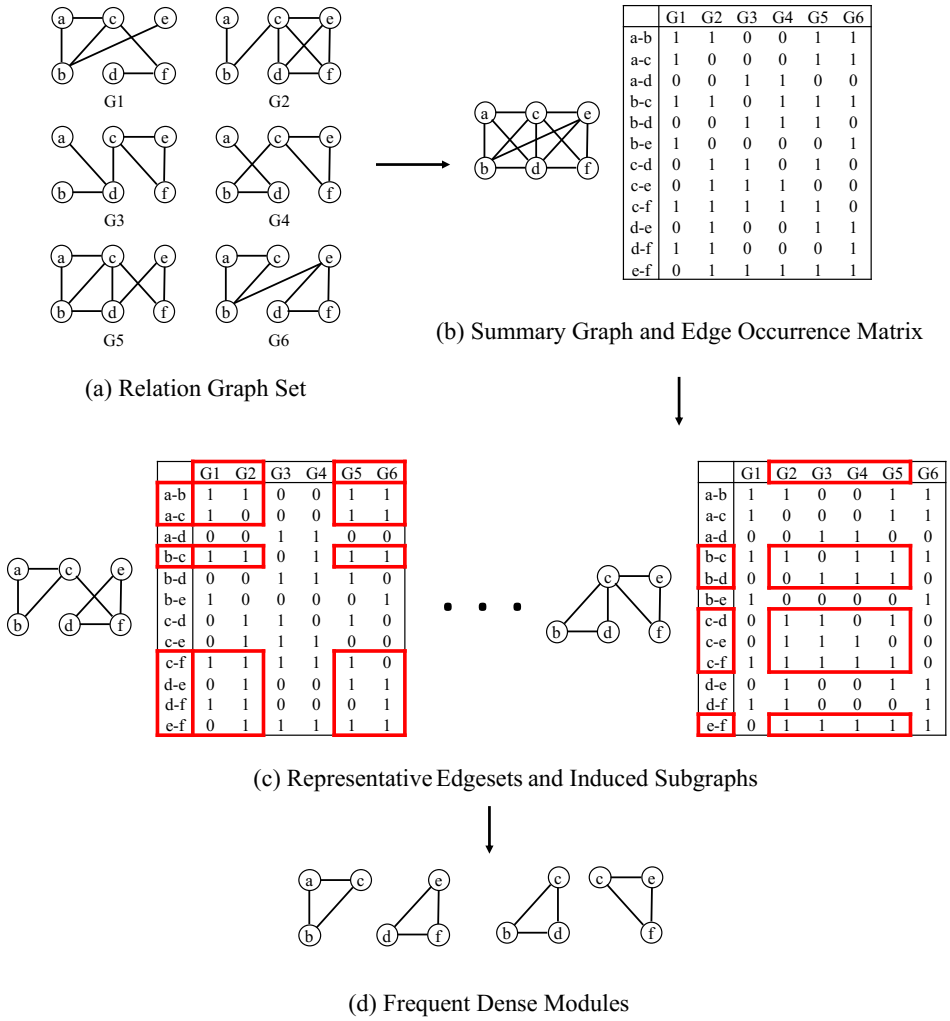
Summary Graph and Edge Occurrence Matrix: Given a relation graph set $\mathcal{G} = \{G_1, G_2, \ldots, G_n\}$ where $G_i = (V, E_i)$, the set of the union of all edges in the graphs is denoted by $E = \{e_1, e_2, \ldots, e_m\} = \bigcup_{i=1}^{n} E_i$. The edge occurrence matrix $\mathcal{B}$ is an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

Figure 1 (b) illustrates the summary graph and the associated binary edge occurrence matrix for the relation graph set in (a). For example, the first row of the edge occurrence matrix shows that the edge (a, b) is present in graphs $\{G1, G2, G5, G6\}$.

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 written as $G[E']$) is a graph whose edgeset is E' and the node set is all the nodes that constitute the endpoints of the edges, i.e., $V' = \bigcup V(e)$ for all $e \in E'$ where $V(e)$ denotes the endpoints of e.

Note that an edge-induced subgraph does not have isolated nodes since each node that is present in the induced subgraph has 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.

A frequent subgraph of a graph set is a subgraph that occurs in at least *minsup* (support threshold) graphs. The supporting graphs



	G1	G2	G3	G4	G5	G6
a-b	1	1	0	0	1	1
a-c	1	0	0	0	1	1
a-d	0	0	1	1	0	0
b-c	1	1	0	1	1	1
b-d	0	0	1	1	1	0
b-e	1	0	0	0	0	1
c-d	0	1	1	0	1	0
c-e	0	1	1	1	0	0
c-f	1	1	1	1	1	0
d-e	0	1	0	0	1	1
d-f	1	1	0	0	0	1
e-f	0	1	1	1	1	1

Figure 1: Steps in mining frequent dense subgraphs: (a) Relation graph set; (b) Summary graph and the binary edge occurrence matrix generated from the relation graph set; (c) Representative frequent edgesets and subgraphs induced by them; (d) Dense modules mined from edge-induced subgraphs

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 for each G in $sup(G', \mathcal{G})$ and k is the number of graphs in which the subgraph appears. When the graph dataset in understood from the context, we refer to $sup(G', \mathcal{G})$ simply as $sup(G')$.

Frequent Subgraph: Given a relation 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 the number of graphs in $sup(G')$ is at least *minsup* graphs, i.e., $|sup(G')| >= minsup$.

The definition of subgraph requires all the edges to appear in the supporting graph. This is a strict requirement and in gene coexpression networks, some links might be dropped due to correlation cutoff or the links might not show strong correlation becuase of experimental noise. Thus we relax the constraints and introduce the approximate frequent subgraph that is a relaxed form of the frequent subgraph by allowing missing edges (noise).

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, the subgraph induced by an edgeset E' is an approximate frequent subgraph if and only if there exists a graph set $D \subseteq \mathcal{G}$ such that $|D| \geq minsup$ and for every edge $e \in E'$, e occurs in at least $\lfloor |D| * (1 - r) \rceil$ graphs in D, the nearest integer to $|D| * (1 - r)$.

To ensure that the subgraph appears in a large enough set of graphs, we require that the subgraph be supported by at least *minsup* graphs. The minimum support threshold *minsup* is essentially the

number of columns in the binary matrix that support the edgetset. Moreover, we only mine large edgeset with at least *minsize* edges.

The noise ratio r is a real number between 0 and 1, which represents how much noise is allowed. An edge e need not be present in every graph in D. For example, the graph in the left side of Figure 1 (c) is an approximate frequent subgraph of the relation graph set in (a) for $minsup = 4$ and $r = 0.25$, because every edge in the graph occurs in at least three out of the four graphs in $\{G1, G2, G5, G6\}$.

The set of all approximate 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 first step of our 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.

Set of Representative Edgesets: Given a set of edgesets $\mathcal{F}$ and edgeset similarity threshold s, a subset $\mathcal{F}' \subseteq \mathcal{F}$ is a set of representative edgesets if for every edgeset $E \in \mathcal{F} \setminus \mathcal{F}'$, there exists an edgeset $E' \in \mathcal{F}'$ such that $sim(E, E') \geq s$, where $sim(E, E')$ is the similarity between the two sets.

We are interested in dense subgraphs in these approximate frequent subgraphs as these edge-induced subgraphs are not necessarily dense.

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. G is dense if its density is greater than or equal to a minimum density threshold.

In this work we mine dense subgraphs from representative frequent subgraphs. We follow a two-step approach to mine approximate frequent dense subgraphs as illustrated in Figure 1. In the first step, we mine a set of representative frequent edgesets using an online data summarization method, as shown in (b-c). In (c), the frequent edgesets pruned in the summarization process are omitted, and only the representative frequent edgesets are shown. In the second step, we mine dense modules from the subgraphs induced by the edgesets, as shown (d). For this step, we use the Dense Module Enumeration (DME) algorithm [4]. We first discuss the method for mining the set of representative frequent edgesets from the binary edge occurrence matrix.

3 MINING REPRESENTATIVE FREQUENT EDGESETS

There are mainly two approaches for mining representative approximate frequent edgesets. The first approach is to mine all frequent edgesets and then cluster these edgesets and choose a representative pattern for each group. The traditional k-medoids algorithm can be employed for clustering. The major challenge with clustering-based approaches is the need to calculate the pairwise distance measure between every pair of frequent edgesets. Given the size of the set of frequent edgesets, this can be computationally intractable. In order to mine a set of representative frequent edgesets from a relation graph set, we employ the modified BiBit algorithm described in [16] and integrate an on-line data summarization method while mining these edgesets. The modified BiBit algorithm is a biclustering algorithm which mines biclusters with high density of ones from a binary matrix. The row set of each bicluster corresponds to a frequent edgeset.

The online data summarization is a data summarization method in which the data is processed as they are produced. In this case, the edgesets are processed as they are mined by the biclustering algorithm. We begin with an empty set of representative edgesets. When an approximate frequent edgeset is found, we check whether it has a similar edgeset (based on a user-defined threshold) in the representative set. If there is no similar edgeset, we add the edgeset to the set of representative edgesets. As a result, the final set contains edgesets such that every edgeset not in the set has at least one similar edgeset in the representative set. Moreover, no two edgesets in the representative set are similar.

3.1 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 set 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 subgraphs is the same as the set of frequent subgraphs. And for the special case when the similarity threshold is set to 0, the first encountered frequent edgeset is the only pattern in the approximate as it is 'similar' to all other edgesets.

3.2 Algorithm

Our algorithm for mining representative frequent edgesets is illustrated in Algorithm 1. The algorithm takes a relation graph set that has the $m \times n$ binary edge occurrence matrix $\mathcal{B}$ in which the rows correspond to edges and the columns correspond to graphs. In the algorithm, $S(i)$ denotes the set of columns (graph) that are set to 1 for row j (edge j), i.e., $S(i) = \{j \mid \mathcal{B}_{ij} = 1\}$, and $S(i, j) = S(i) \cap S(j)$ denotes the set of columns that are set to 1 for both edges i and j. For example, in the matrix in Figure 1 (b), $S(1) = \{1, 2, 5, 6\}$. The algorithm selects a pair of rows i and j and generates the bit-pattern $(\{i, j\}, S(i, j))$, which is a tuple of two rows and its supporting columns. The bit-pattern $(\{i, j\}, S(i, j))$ is used as a seed for a bicluster if $|S(i, j)| \geq minsup$, and $S(i, j)$ represents the column set for the bicluster. Only edgepairs that appear in at least *minsup* graphs are extended (line 4). The algorithm extends each edge pairs with edges that can be added without violating

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 Representative Frequent Edgesets     */
1  F ← ∅
2  for every edge pair (i, j) ∈ E do
3      S(i, j) = S(i) ∩ S(j)                  // common graphs
4      if S(i, j) is new and |S(i, j)| ≥ minsup then
5          I ← {i, j}
6          for every remainder edge, q ∈ E \ I do
7              if |S(q) ∩ S(i, j)|/|S(i, j)| ≥ 1 − r then
8                  I = I ∪ {q}
9          if |I| < minsize then
10             continue
11         similar ← FALSE
12         for every edgeset I' ∈ F do
13             if sim(I, I') ≥ s then
14                 similar ← TRUE
15                 break
16         if similar is FALSE then
17             F = F ∪ I
   /* Extracting Dense Modules                    */
18 X ← ∅
19 foreach frequent edgeset F_i ∈ F do
20     g_i = G[F_i]                  // edge Induce Subgraph
21     X = X ∪ DME(g_i, α)
22 return X
```

the noise threshold. Each remaining row q is added to the bicluster if $|S(q) \cap S(i,j)|/|S(i,j)| \geq 1-r$, that is, if $S(q)$ contains some entries of $S(i,j)$ in such a way that the noise constraint is not violated (lines 6-8). The result is a bicluster with density greater than or equal to $1-r$. The row set of the bicluster represents a frequent edgeset. If the number of rows for a bicluster is less than the minimum size threshold, the bicluster is not added to the result (lines 9-10). Before adding the bicluster to the set of representative frequent edgesets $\mathcal{X}$ (lines 16-17), the algorithm ensures that no similar edgeset is already in the representative set (lines 12-15). Finally dense modules are extracted from the edge-induced subgraph of the summary graph for each representative frequent edgeset (lines 18-21).

4 EXPERIMENTS

To evaluate the effectiveness of our method, we mined the set of representative approximate frequent edgesets and the associated dense modules from 35 tissue gene coexpression networks constructed by the Genetic Network Analysis Tool [12]. The gene coexpression networks were constructed from Genotype-Tissue Expression (GTEx) data [1]. Each coexpression network is constructed from the gene expression of non-diseased tissue samples. On average, each coexpression network contains 9,998 genes and 14,415 links. There are total of 1,548,622 unique links that appear in at least one network and 4,127 edges that appear in at least 20 networks, and each link appears in 3.28 networks on average.

4.1 Effect of Data Summarization

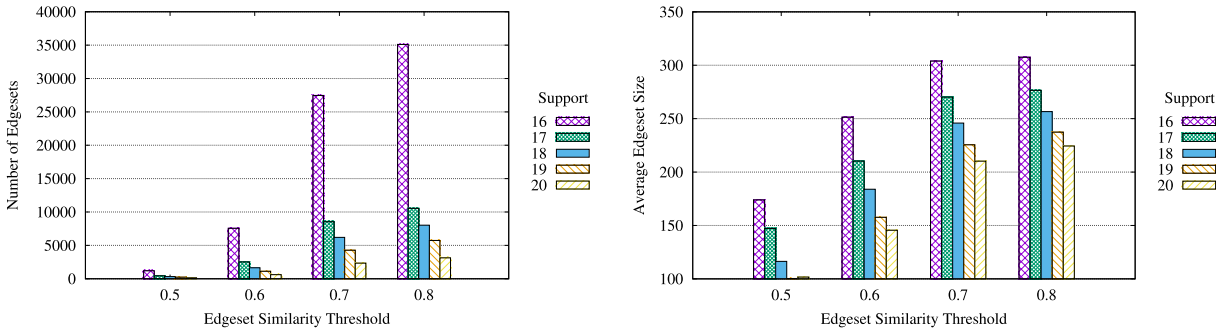
To evaluate the effectiveness of the proposed approach, we ran the algorithm on the binary edge occurrence matrix constructed from the 35 gene coexpression networks, for support threshold $minsup \in \{16, 17, 18, 19, 20\}$, noise threshold $r = 0.1$, and edgeset similarity threshold $s \in \{0.5, 0.6, 0.7, 0.8\}$. Figure 2 (a) shows how the number of frequent edgesets varies for different edgeset similarity threshold values. We can see that the number of frequent edgesets decreases with increasing support threshold and increases with increasing the edgeset similarity threshold. This is expected because less number of representative edgesets is needed for lower similarity threshold. Figure 2 (b) shows how the average edgeset size varies for different edgeset similarity threshold values. We see that the average edgeset size increases with increasing edgeset similarity threshold.

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 ($s = 0.3$)	17	13	14	16
With summarization ($s = 0.4$)	61	38	31	38
With summarization ($s = 0.5$)	215	141	127	145
With summarization ($s = 0.6$)	579	613	599	826
With summarization ($s = 0.7$)	1,546	2,341	2,569	2,993
With summarization ($s = 0.8$)	2,789	3,138	3,221	3,244

To evaluate the effect of online frequent edgeset summarization, we mined approximate frequent edgesets and representative frequent edgesets for $minsup = 20$ We used edgeset similarity thresholds 0.3 to 0.8 for mining representative approximate frequent edgesets. Tables 1 shows the reported number of frequent edgesets for various similarity thresholds. The number of representative frequent edgeset increases as we increase the similarity thresholds. For a small similarity threshold, a small number of edgesets can claim to represent the entire set of approximate frequent edgesets. And for a large similarity threshold, fewer edgesets are similar to each other and thus the number of representative patterns is larger.

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



(a) Number ofEdgesets vs. Similarity Threshold

(b) AverageEdgeset Size vs. Similarity Threshold

Figure 2: Number of frequent edgesets and average edgeset size for varying edgeset similarity threshold values

4.2 Topological Analysis of Frequent Edgesets and Frequent Dense Modules

We ran the algorithm on the binary edge occurrence matrix for support threshold $minsup \in \{16, 17, 18, 19, 20\}$, noise threshold $r \in \{0, 0.1, 0.2, 0.3\}$, and edgeset similarity threshold $s = 0.6$. Figure 3 shows how the number of frequent edgesets and the average edgeset size vary for different noise threshold values. We see that the number of frequent edgesets and the average edgeset size both increase with increasing noise because for larger noise. For a large noise, a seed edgepair has more candidate edges that can be added without violating the noise threshold.

We then mined dense modules from the subgraphs induced by the frequent edgesets, using the DME algorithm [4], with density thresholds 0.5 and 0.6, and only modules of size four or larger were considered. Table 2 shows the topological properties of the frequent dense modules for $minsup = 17, 18, 19, 20$, noise threshold $r = 0, 0.1, 0.2$, and edgeset similarity threshold $s = 0.6$. M' denotes the number of frequent edgesets that have at least one dense module for the specified density threshold, $\overline{DM}$ denotes the average number of dense modules in the edge-induced subgraph of each edgeset, and $\overline{V'}$ denotes the average size of the dense modules. It shows that the number of edgesets with at least one dense module and the average number of dense modules both decrease as the support threshold is increased.

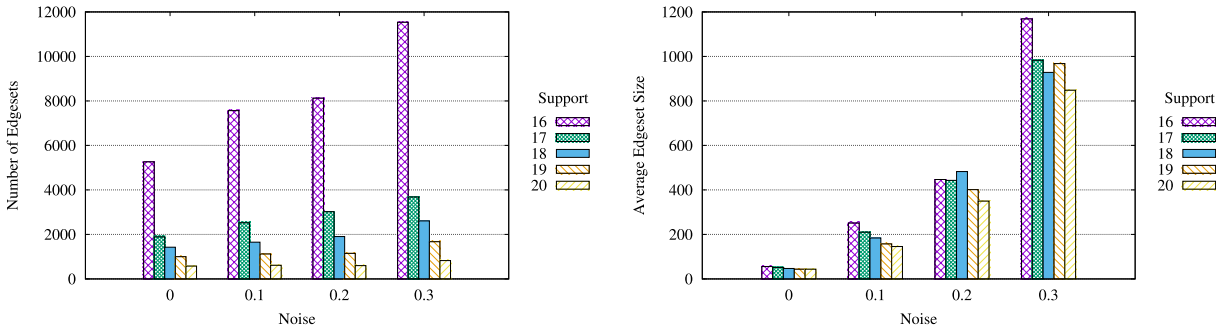
4.3 Gene Ontology Enrichment Analysis

To assess the biological significance of the reported modules, we conducted Gene Ontology enrichment analysis of the reported unique frequent dense modules. The analysis shows that the modules are enriched with KEGG pathways and molecular functions. A frequent dense 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. E_M and E_K denote the percent enriched in GO molecular functions and KEGG pathways respectively. The results show that frequent dense modules with smaller noise ratios are more likely to be enriched. The GO molecular functions have higher enrichment than KEGG pathways since there are much more molecular functions than KEGG pathways and they cover more genes from the graph dataset. The set of genes in a frequent dense module can be enriched with multiple biological annotations. Also, 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 $density = 0.5$; count indicates the number of frequent dense modules in which the annotation is enriched.

Figure 4 shows an example of an approximate frequent edgeset for $sup = 17$, $noise = 0.1$. (a) show the submatrix of the edge occurrence matrix that shows the occurrences of edges of the edgeset in the 35 networks. The rows correspond to the edges in the edgeset, and the columns correspond to coexpression networks. (b) shows the dense modules mined from the subgraph induced by the edgeset, using density 0.5. Nodes are labeled by their corresponding gene identifiers. The genes in this representative approximate edgeset are enriched with five KEGG pathways: OXIDATIVE_PHOSPHORYLATION, CARDIAC_MUSCLE_CONTRACTION, ALZHEIMERS_DISEASE, PARKINSONS_DISEASE, and HUNTINGTONS_DISEASE. Moreover, two Gene Ontology terms were enriched in this edgeset: ELECTRON_TRANSFER_ACTIVITY, and OXIDOREDUCTASE_ACTIVITY.

5 CONCLUSION

Mining gene modules that are recurrent in multiple gene coexpression networks has applications in functional gene annotation and biomarker discovery. We have proposed a two-step algorithm to mine frequent dense modules from a set of multiple coexpression networks. First, we mine a set of representative frequent edgesets from the binary edge occurrence matrix constructed from the set of coexpression networks, using an online data summarization method. Second, dense modules are extracted from the subgraphs induced by the frequent edgesets. The key contribution of this work is that by mining representative edgesets, we addressed the problem of the large number of edgesets being reported and the

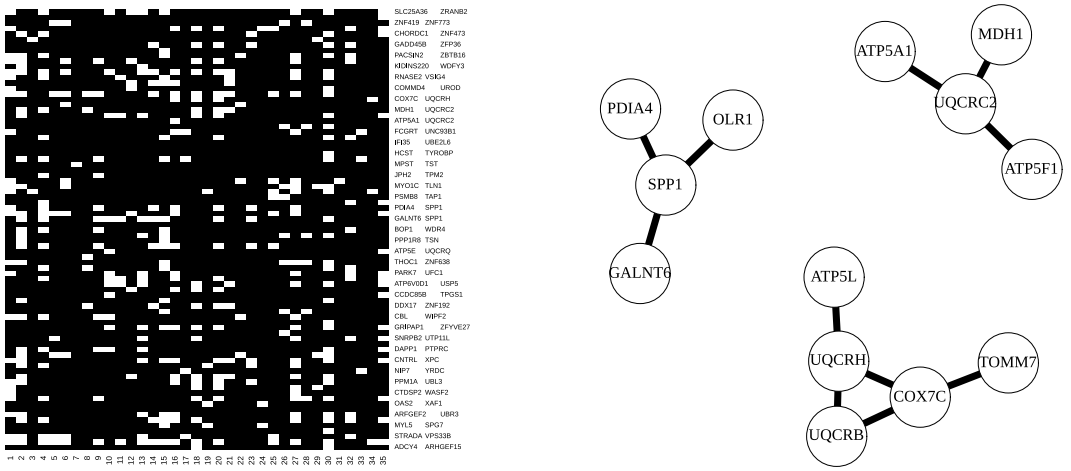


(a) Number of Edgesets vs. Noise Ratio (b) AverageEdgeset Size vs. Noise Ratio

Figure 3: Number of frequent edgesets and average edgeset size for varying noise ratio values

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	513	11.2	4	2.3 K	51.8	4.2	2.9 K	206.7	4.3
	0.6	20	1.2	4	646	4.7	4.2	1.7 K	15.2	4.4
18	0.5	346	10.6	4	1.4 K	42.8	4.1	1.8 K	252	4.3
	0.6	6	1.2	4.2	362	4	4.2	1.1 K	18.9	4.4
19	0.5	238	9.3	4	941	32.7	4.1	1.1 K	187.8	4.3
	0.6	6	1	4	190	3.2	4.2	579	14.6	4.4
20	0.5	134	9.9	4	499	29.5	4.1	540	153.6	4.2
	0.6	3	1	4	84	3.3	4.2	265	13.1	4.3



(a) Submatrix for Frequent Edgeset (b) Frequent Dense Modules

Figure 4: Sample frequent edgeset for $minsup = 17$ and $noise = 0.1$, and dense modules in the edgeset for $density = 0.5$

high overlap between the edgesets. As a result, the analysis is computationally less intensive and the redundancy between the repoted modules in reduced. Experiments on human gene coexpression networks show that the reported modules are enriched with known GO molecular functions and KEGG pathways.

ACKNOWLEDGMENTS

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

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	80.4	64.3	65.5	55.7	51	43.4
	0.6	90.9	45.5	80.5	60.3	68.9	49.8
18	0.5	81.6	62.6	71.1	59.6	51.3	43
	0.6	100	50	83.9	56.7	68.5	49.5
19	0.5	87.2	66.5	75	61.9	55.2	46.7
	0.6	100	50	83.8	54.1	71.7	53.3
20	0.5	85.8	67.1	77.5	66.5	61.6	52
	0.6	100	33.3	91.2	66.7	76.4	56.1

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	1996
Rrna Binding	503
5s Rrna Binding	277
Electron Transfer Activity	271
Oxidoreductase Activity Acting On Nad P H	214
Nadh Dehydrogenase Activity	208
Antigen Binding	194
Immunoglobulin Receptor Binding	175

KEGG Pathway	Count
Ribosome	2001
Huntingtons Disease	503
Oxidative Phosphorylation	493
Parkinsons Disease	472
Alzheimers Disease	464
Cardiac Muscle Contraction	240
Autoimmune Thyroid Disease	58
Mapk Signaling Pathway	54
Aminoacyl Trna Biosynthesis	52
Protein Export	41

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] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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–.

[9] 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.

[10] 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.

[11] 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.

[12] 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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.