SiS: Significant Subnetworks in Massive Number of Network Topologies

Md Mahmudul Hasan, Yusuf Kavurucu and Tamer Kahveci
Department of Computer and Information Science and Engineering,
University of Florida, Gainesville, Florida 32611, USA.
{*mmhasan, yusuf, tamer*}*@cise.ufl.edu*

***Abstract*—Availability of abundant biological network data and its noisy nature necessitates extracting reliable subnetworks hidden in it. One key step towards that goal is to discover the subnetworks that appear frequently in a collection of networks . This paper presents a method, named SiS (Significant Subnetworks), to discover most probable subnetworks (i.e., subnetworks that have the highest chance to exist) in a large collection of biological networks where each node is labeled with the corresponding molecule (such as gene or protein). SiS builds a template network which summarizes the entire set of input networks. It then grows subnetworks that are most probable with the guidance of this template network. Our experiments demonstrate that our method scales to very large datasets and subnetworks easily. On the metabolic networks of the eukaryote organisms, our method runs from a few seconds to a few minutes depending on the subnetwork size. MULE, an existing method for the same problem, takes hours or does not complete for days on the same dataset. Our results also suggest that the most probable subnetworks are often the most frequent ones as well.** [1]

***Keywords*-frequent subnetworks; biological networks.**

I. Introduction

One of the fundamental goals of molecular biology is to understand the biological processes that are the driving forces behind organisms' functions [1]. Studying biological networks provides significant insight towards this goal.

A fundamental challenge in studying biological networks is that they can be noisy. One major source of noise is the errors in the measurements of high-throughput experiments. As a result, multiple high-throughput experiments among the same set of molecules can lead to different sets of inferred interactions. Another source of noise arises from the mathematical models used to build biological networks [2]; there can be alternative network topologies that explain the high-throughput experiments equally well under the same mathematical model [3]. *Regardless of the source of noise, one way to resolve the differences among alternative network topologies is to identify a reliable backbone of the network that exists in a significant number of these alternative networks.*

Biological networks are often modeled as graphs in which nodes correspond to molecules and edges correspond to the interactions between them. Using a graph model, we denote a network with $G = (V, E)$ where V and E denote the set of nodes (e.g., genes or proteins) and directed edges (i.e., interactions) respectively. Undirected interactions, such as protein-protein interactions, are a special case of this model and have practically no difference from the directed ones from the point of view of this paper. Using this representation, following defines the goal of this paper briefly. We provide a formal problem definition in Section II after presenting essential definitions and terms.

Goal statement*: Assume that we are given a collection of m alternative biological networks each built on the same set of molecules and denoted by graph $G_i = (V, E_i), i \in \{1, 2, \ldots m\}$. Given positive integers n and k, we would like to find the n connected subnetworks each containing k edges that appear in the largest number of these networks.*

The problem described above has deep connections with frequent subgraph discovery [4] and network motif identification [5] problems. Both of these problems are computationally hard as many solutions to them require solving subgraph or graph isomorphism problems which are NP-Complete [6] and GI-Complete [7] respectively.

The cost of searching for frequent subgraphs exhaustively increases exponentially with growing size of input graph and subgraph. Therefore, it is not feasible to discover frequent subgraphs by exhaustively enumerating all possible subgraphs in the input graphs. As a result, a number of heuristics have been developed to avoid impractical execution time. However, these methods have several drawbacks. Briefly, some of them ignore the contents of the nodes and entirely focus on network topology [8]; some do not scale to large networks and subnetworks [9] or can only find subgraphs in a single network [10].

Contributions. In this paper, we propose a new method, named *SiS* (**Si**gnificant **S**ubnetworks), that efficiently discovers subnetworks that appear frequently in a set of biological networks. Given a large collection of biological networks with alternate topologies, our method starts by constructing a new network, namely the *template network*. This network is the union of the interactions of all the networks in the input dataset. We assign weight to each interaction in the template network as the negative logarithm of the probability that that interaction exists in a randomly selected network in the input dataset. Our algorithm then grows a subnetwork in the template network with the guidance of these weights. Given positive integers n and k, it seeks the top n connected subnetworks each containing k interactions and having the

[1]This work was supported partially by NSF under grants CCF-0829867 and IIS-0845439.

978-1-4673-2560-8/12/$31.00 ©2012 IEEE

smallest total edge weight. In other words, it looks for those subnetworks with the highest chance of appearing in a randomly drawn network from the input dataset. We compute upper and lower bounds to the total edge weights of the top n subnetworks. We use these bounds to filter massive number of unpromising subnetworks quickly. We also update the upper bound dynamically to have tighter bounds as we search the template network.

Our results demonstrate that our method scales to very large datasets (i.e., datasets with over a thousand networks, having a total of hundreds of thousands of nodes and edges) easily. For instance, it takes our method less than five minutes to find the most frequent subnetwork with up to 31 interactions in metabolic networks of 145 eukaryotes. MULE [11], one of the state of the art methods, takes around 7.5 hours to find the same solution. Using our method, we observe that eukaryotes are more conserved than prokaryotes. We also observe that the prokaryote networks often contain the conserved subnetworks in eukaryotes. However the opposite does not hold for a substantial number of frequent subnetworks in prokaryotes.

The organization of the rest of this paper is as follows: Section II formally defines the problem and describes the proposed solution. Section III evaluates our method experimentally. We conclude briefly in Section IV.

II. Proposed Method

In this section, we first define the problem formally and present some of the common terminology used in this paper (Section II-A). We then describe the technical details of our method (Section II-B).

A. Preliminaries and problem description

We start by defining the terminology needed to understand the rest of this paper.

Definition 1 (Graph)**.** *A graph $G = (V, E)$ consists of a set of nodes V and a set of edges $E \subseteq (V \times V)$. An edge $e = (u, v) \in E$ starts from node u (source) and ends at node v (target).*

We say that nodes u and v are *adjacent* to each other in G if there is an edge (u, v) or (v, u) between them in G. In that case, we also say that nodes u and v are *incident* to the edge (u, v) or (v, u).

Definition 2 (Subgraph)**.** *A subgraph of a graph $G = (V, E)$ is a graph $G' = (V', E')$ where $V' \subseteq V$ and $E' \subseteq (V' \times V') \cap E$.*

To simplify our terminology, we use the terms graph and subgraph to represent network and subnetwork, respectively in the rest of this paper.

Definition 3 (Path)**.** *Let $G = (V, E)$ be a graph. Assume that $v_i \in V$ for $i = 0, 1, ..., t$. We say that there is a path from v_0 to v_t if v_i and v_{i+1} are adjacent to each other for $i = 0, 1, ..., t - 1$.*

Definition 4 (Connected Graph)**.** *A graph $G = (V, E)$ is connected if there is a path between any pair of nodes $u, v \in V$.*

In this paper, we consider only connected subgraphs. Given a positive integer k, we use the term *size-k* subgraph to denote a connected subgraph with k edges. We express the concept of existence of a subgraph G' in a graph G using the following indicator function:

$$exists(G', G) = \begin{cases} 1 & \text{if } G' \text{ is a subgraph of } G \\ 0 & \text{otherwise.} \end{cases}$$

Assume that we are given a collection $\mathcal{D}$ of m graphs represented as $\mathcal{D} = \{G_1, G_2, ..., G_m\}$. Here, all the G_is have the same set of labeled nodes, V. Hence, we represent each graph as G_i = (V, E_i) where $E_i \subseteq (V \times V)$. The frequency of a subgraph G' over $\mathcal{D}$ denotes the number of graphs $G_i \in \mathcal{D}$ that contains G'. The following function computes this:

$$frequency(G', \mathcal{D}) = \sum_{G_i \in \mathcal{D}} exists(G', G_i)$$

Now, we are ready to formally define the concept of most frequent subgraph.

Definition 5 (Most Frequent Size-k Subgraph)**.** *Let $\mathcal{D}$ be a collection of graphs built over the same set of nodes and k be a positive integer. The most frequent size-k subgraph in $\mathcal{D}$ is a subgraph G' with k edges for which $frequency(G', \mathcal{D})$ is maximum among all size-k subgraphs.*

A simple extension to Definition 5 takes one more input parameter, a positive integer (n), and returns the n most frequent size-k subgraphs in $\mathcal{D}$.

B. SiS: Significant Subnetworks

In this section, we define the terminology relevant to our method first. We then describe the proposed method.

We define the frequency of an edge, e in $\mathcal{D}$ (denoted with $frequency(e, \mathcal{D})$) as the number of $G_i \in \mathcal{D}$ that contain e. We normalize the frequency of each edge to compute another function denoted by $fr(e, \mathcal{D})$ as $fr(e, \mathcal{D}) = \frac{frequency(e, \mathcal{D})}{|\mathcal{D}|} = \frac{frequency(e, \mathcal{D})}{m}$.

Notice that the function $fr(e, \mathcal{D})$ is equal to the probability that a randomly selected graph in $\mathcal{D}$ contains e. Similarly, for any collection of edges $\{e_1, e_2, ..., e_k\}$ from the graphs in $\mathcal{D}$, the probability that a randomly drawn graph from $\mathcal{D}$ contains all of these edges is $\Pi_{i=1}^{k} fr(e_i, \mathcal{D})$. Using this observation, next, we introduce a special graph called the *template graph* that will help us in building our method.

Definition 6 (Template Graph)**.** *Let $\mathcal{D}$ be a collection of graphs G_i with $G_i = (V, E_i)$. The template graph of $\mathcal{D}$ is an edge weighted graph $T = (V, E_T, \psi())$, where $E_T = \bigcup_i E_i$ and $\psi()$ is a function $\psi : E_T \rightarrow \mathbb{R}$ such that $\forall e \in E_T, \psi(e) = -log(fr(e, \mathcal{D}))$.*

Intuitively, the template graph summarizes the frequencies of all edges in $\mathcal{D}$. From Definition 5, we know that a frequent size-k subgraph G' appears in many graphs in $\mathcal{D}$. As a result, we conjecture that it consists of frequent edges of T (i.e., edges that have large $fr(e, \mathcal{D})$ values and thus, small $\psi(e)$ values). However, the converse is not always true. This leads to our next conjecture that the probability that such a frequent G' appears in a randomly selected graph in $\mathcal{D}$ tends to be higher than infrequent subgraphs of the same size. To exploit these conjectures, we define the following score function for G':

$$score(G', \mathcal{D}) = \sum_{e \in G'} \psi(e) \quad (1)$$

Clearly, the larger the probability $\Pi_{e \in G'} fr(e, \mathcal{D})$ is, the smaller the value of $score(G', \mathcal{D})$ is. This holds because of the following:

$$score(G', \mathcal{D}) = -\sum_{e \in G'} \log(fr(e, \mathcal{D})) = -\log(\Pi_{e \in G'} fr(e, \mathcal{D}))$$

The purpose of introducing the score function, as we describe later in this section, is to transform a multiplicative expression of the probability into an additive one which helps us develop our method. Following definition summarizes the discussion above.

Definition 7 (Most Probable size-k Subgraph)**.** *Let $\mathcal{D}$ be a collection of graphs built over the same set of nodes and k be a positive integer. The most probable size-k subgraph in $\mathcal{D}$ is a subgraph G' with k edges with the smallest $score(G', \mathcal{D})$ value as compared to other size-k subgraphs in $\mathcal{D}$.*

We develop a heuristic algorithm, called ***Si***gnificant ***S***ubnetworks (SiS), that discovers the n most probable size-k subnetworks very quickly where positive integers n and k are user supplied parameters. As we demonstrate in our experiments, most probable subnetworks tend to be frequent as well in practice.

Although Definitions 5 and 7 are not identical, we observe that they return identical or nearly identical results (details in Section III). Our method can identify most probable subgraphs in seconds or minutes on a standard desktop for very large datasets or subgraphs while it takes many hours or days (or becomes impossible) to find the most frequent ones using existing methods (See Section III-B). Briefly, our algorithm explores the size-k subgraphs in the template graph. It seeks those subgraphs with small score values (i.e., those with a high chance of being frequent). As our algorithm searches the template graph, it computes *lower* and *upper bound* values to $score(G', \mathcal{D})$ for size-k' subgraphs G's ($k' = 1 \dots k$). It uses these bounds to prune the search space. It then updates upper bounds with better values until it prunes or explores the entire set of possible size-k subgraphs. Next, we elaborate on different steps of SiS.

Lower bound calculation*:* We calculate the minimum value of $score(G', \mathcal{D})$ for any subgraph G' with k' edges. We get a lower bound to that value by considering the most frequent edges of T which are connected together (i.e., edges with smallest $\psi()$ values). Finding such a connected subgraph is itself a difficult problem as the number of possible size-k' subgraphs grow rapidly with increasing value of k'. Instead, we quickly obtain a lower bound to this value by dropping the constraint that the subgraph is connected. This relaxation allows us to pick the k' edges of T with the lowest $\psi()$ values. The sum of the weights of these edges is guaranteed to be less than or equal to the score of any size-k' subgraph of T.

Upper bound calculation*:* We employ a greedy approach to calculate an upper bound to $score(G', \mathcal{D})$ for any size-k' subgraph G' in T as follows. We start from an arbitrary edge $e \in E_T$ and initialize G' to only that edge. We grow G' to a size-k' subgraph by adding one edge at a time. We always choose the edge that is adjacent to the current subgraph G' and has the lowest $\psi()$ value among all such edges for that purpose. Notice that the score of the resulting G' depends on the initial edge e we started with. We repeat this process for each edge in E_T by using it as the starting edge and compute the score values of the resulting size-k' subgraphs. Among all the resulting $|E_T|$ score values for all possible starting edges, we use the smallest one as the upper bound. We can prove by contradiction that any subgraph G' generated this way has a score value at least as much as the optimal one. Briefly, a connected size-k' subgraph either has the smallest score value among all size-k' subgraphs (i.e., it is optimal) or it is larger than the smallest one (i.e., it is not optimal).

Exploring the search space*:* Before we start searching for the most probable subgraphs, we calculate and store the lower and upper bound values for all $k' = \{1 \dots k\}$. We store these numbers in arrays $LB[1 \dots k]$ and $UB[1 \dots k]$, respectively. Here, the i-th entry of these arrays is the bound for size-i subgraphs. We first describe how we explore the subgraphs. Later, we discuss how we use the lower and upper bounds to accelerate the search process.

We explore the search space by growing subgraphs from smaller ones as follows. Let us denote the subgraph considered at any point during the search with $G' = (V', E')$. We set G' to a size-1 subgraph by selecting a single edge, e_i from the template graph (i.e., $E' = \{e_i\}$). We then grow G' by inserting a new edge e_j into E' if that edge satisfies all of the following three constraints (below we denote the subgraph that is obtained after adding e_j into E' with G''):

(i) e_j is incident to v such that $v \in V'$.

(ii) The index of the new edge (i.e., j) is greater than the index of the first edge in E' (i.e., i).

(iii) $score(G'', \mathcal{D}) + LB[k - |E'| - 1] \leq UB[k]$

Here, the first constraint ensures that the subgraph we build remains connected. The second one ensures that we enumerate the search space in a non-redundant way. That

is, we never construct the same subgraph twice. The last one filters a large set of unpromising subgraphs as soon as the lower and upper bounds prove that they cannot have the optimal score value. We repeat the above described search process iteratively by initializing the size-1 subgraph to all possible edges in the template graph one by one.

Notice that, while growing the subgraph, if an edge satisfies all the constraints, then it leads to a new subgraph that has one more edge than the previous one. Assume that this new subgraph has k' edges. The score of this subgraph is an upper bound to that of the optimal size-k' subgraph in T. This follows from the fact that the optimal size-k' has the smallest score among all subgraphs of the same size. From this observation, we update the upper bound value $UB[k']$ with the smaller of $UB[k']$ and the score of the new subgraph. Thus, we obtain tighter upper bound values as we explore the search space leading to further pruning of the search space. As we later show in our experiments, this makes SiS scale to very large datasets and subgraph sizes for which existing methods fail.

Finally, it is worth mentioning that the method described above searches the subgraph with the smallest score. Extending it to find the n subgraphs with the smallest scores is trivial. The primary difference is that we compute upper and lower bounds to the top n' size-k' subgraphs for $k' = 1, \ldots, k$ and $n' = 1, \ldots, n$.

Post processing and extension*:* Once SiS finds the n most probable subgraphs, it computes their frequencies. This is computationally cheap as it requires checking if a labeled subgraph is contained in a labeled graph for all pairs of subgraphs and database graphs. SiS then returns the top-n most probable subgraphs along with their actual frequencies in $\mathcal{D}$.

We say that a frequent subgraph $G' = (V', E')$ is *m*aximal if there is no other subgraph $G'' = (V'', E'')$ such that $V' \subseteq V''$, $E' \subseteq E''$ and frequency $(G'', \mathcal{D})$ is greater than a user supplied frequency cutoff. One possible extension to SiS is its ability to find maximal frequent subgraphs rather than frequent size-k subgraphs for a fixed value of k. We do this in the following manner. We generate n most frequent size-k subgraphs for moderately large k values. Then for each of them, we add one edge at a time such that the new edge is connected to the existing subgraph and the frequency of the resulting larger subgraph is above the given threshold frequency. In Section III, we report the performance of SiS with this variation.

III. Experiments

In this section, we evaluate the performance of our method. We implemented our method in *C++* language. We performed all the experiments on a Windows 7 desktop computer (Intel Core i7 2.67 GHz CPU, 4 GB of RAM). We describe the datasets below.

Dataset*:* We use metabolic networks of all the eukaryotes and prokaryotes from the KEGG database [12]. A directed edge (u, v) means that an enzyme denoted by u catalyzes a reaction whose product is consumed by a reaction that an enzyme denoted by v catalyzes. Note that an enzyme can catalyze multiple reactions in a pathway; nonetheless, we represent it by a single node in this model. We call this dataset the *global network* dataset as it contains entire metabolic networks of eukaryotes and prokaryotes. We also created two smaller datasets by considering only the enzymes in the pyrimidine network and alanine, aspartate and glutamate network of eukaryotes and prokaryotes. Note that different organisms may have slightly different enzyme sets in the same metabolic network. To ensure that each resulting network in a given dataset has the same set of nodes, we set the node set to the union of all the enzymes in all organisms for that dataset. Table I lists the detailed statistics on the datasets.

Table I
AN OVERVIEW OF THE DATASETS USED IN THE EXPERIMENTS (G: GLOBAL, AAG: ALANINE, ASPARTATE AND GLUTAMATE, P: PYRIMIDINE).

Dataset	Number of Organisms	Entire Dataset		Template Graph	
		Nodes	Edges	Nodes	Edges
Eukaryote-G	145	45,315	78,499	1,413	1,541
Prokaryote-G	1,486	393,681	616,546	1,413	1,676
Eukaryote-AAG	145	2,048	2,951	43	54
Prokaryote-AAG	1,442	20,692	27,334	43	79
Eukaryote-P	145	2,942	5,757	63	103
Prokaryote-P	1,486	31,130	60,802	63	114

A. Subgraphs in datasets

We performed the first set of experiments on the global network datasets. Our goal in these experiments is to find out how conserved the metabolic networks of the organisms in the two branches of life, namely eukaryotes and prokaryotes, are and how well they relate to each other.

We searched for top 50 most probable size-k subgraphs for each value of $k = 6, 7, \ldots 20$ (in total, 750 (50 $\times$ (20 - 5)) subgraphs) separately for Eukaryote-G and Prokaryote-G datasets. Figure 1 (left) plots the result. Our results demonstrate that both eukaryotes and prokaryotes have large and highly conserved subgraphs. Thus, the metabolisms of the organisms in each of these major clades serve same complex functions through the same topology of interactions most of the time. Another important observation is that the metabolic networks of eukaryotes are more conserved than that of prokaryotes even for small subgraphs. This is indeed justified in the literature as functional conservation is often observed in eukaryotic organisms both at the protein level as well as the network level [13]. Finally, as the subgraph size increases, the frequency in both clades decreases. However, this drop is gradual. This implies that if a (potentially large) subgraph is common to many organisms, the probability that they contain more common interactions in addition to those in that subgraph is high. Otherwise, we would have observed an exponentially decreasing frequency value with increasing subgraph size.

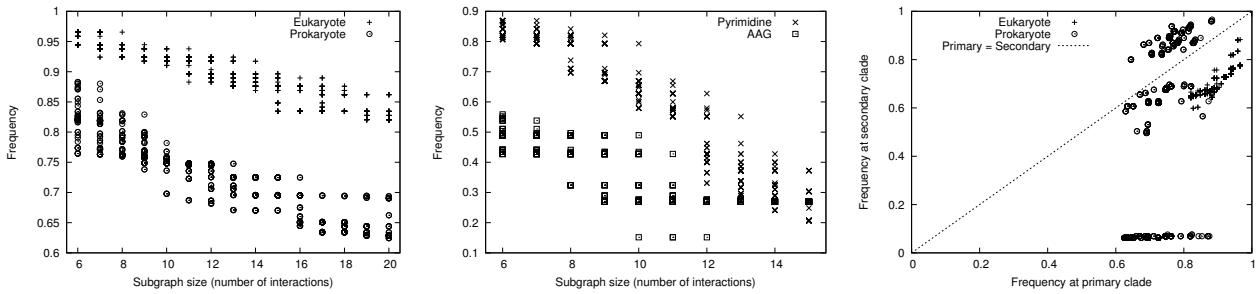

Figure 1. **Left:** Frequency of the top 50 frequent size-k ($k = 6 \ldots 20$) subgraphs in eukaryotes and prokaryotes (global map metabolic pathways). **Center:** Frequency of the top 50 frequent size-k ($k = 6 \ldots 15$) subgraphs in eukaryotes (pyrimidine (P) and alanine, aspartate and glutamate (AAG) metabolic pathways). **Right:** Frequency of the top 50 frequent size-k ($k = 6 \ldots 20$) subgraphs discovered in primary clade (eukaryotes/prokaryotes) and their corresponding secondary clade (prokaryotes/eukaryotes) (global map metabolic pathways).

Based on the above observation, we explored the metabolic networks of eukaryotes and prokaryotes further. This time, we ran our experiments on the alanine, aspartate and glutamate (AAG) and pyrimidine networks (P). We queried top 50 most probable size-k ($k = 6 \ldots 15$) subgraphs in these datasets. Figure 1 (center) plots the frequency values of the subgraphs we found using SiS for eukaryotes (using Eukaryote-AAG and Eukaryote-P datasets). Our results show that the pyrimidine network is significantly more conserved than the alanine, aspartate, and glutamate network. This is more evident for moderate size of k ($k \leq 12$). As subgraph size increases, frequency values for frequent subgraphs start to decrease in both of these subgraphs. We made similar observation for prokaryotes in Prokaryote-AAG and Prokaryote-P datasets.

We, next, evaluated the correlation between the frequent subgraphs in the two major clades (i.e., eukaryotes and prokaryotes). We did this as follows. We first found the 50 most probable size-k subgraphs for each value of $k = 6 \ldots 20$ in each clade using SiS on the global networks (i.e., Eukaryote-G and Prokaryote-G datasets). We then computed the frequency of each of these subgraphs both in the dataset it was found (we call this dataset the *primary clade*) as well as the other dataset (we call this the *secondary clade*). Thus, for each resulting subgraph, we computed two frequency values. Figure 1 (right) plots these frequency values. We observe that the frequent subgraphs in eukaryotes are also frequent in prokaryotes. However, the inverse is not true. There are many frequent subgraphs in prokaryotes that do not exist in most of the eukaryotes. Our observation conforms to the existing biological knowledge that almost all metabolic pathways that exist in eukaryotes also exist in prokaryotes [14]. However, some pathways are only found in prokaryotes, especially in bacteria and archaea living in extreme environments. For instance, prokaryotes can carry out anaerobic respiration through pathways that do not exist in eukaryotes [15]. Also, a number of prokaryotes can produce methane through metanogenesis which deviate them from eukaryotes [16]. Next, we focus on some of those subgraphs.

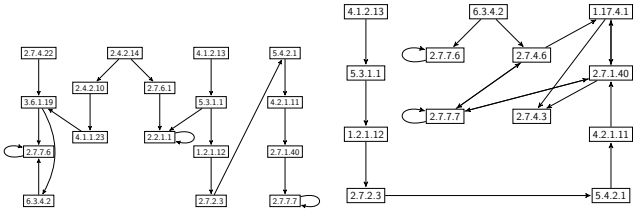

Figure 2. **Left:** A size-20 subgraph which is highly frequent in prokaryotes, but infrequent in eukaryotes. **Right:** A size-20 subgraph which is frequent in both eukaryotes and prokaryotes.

Figure 2 (left) shows a size-20 subgraph that has high frequency (69%) in prokaryotes but very low (only 7%) in eukaryotes. The node labels refer to the Enzyme Commission (EC) numbers of the corresponding enzymes. The subgraph in Figure 2 (right) on the other hand is frequent in both eukaryotes (86%) and prokaryotes (66%). Interestingly, there are 10 common enzymes in these two subgraphs. This is a substantial overlap as the two subgraphs contain only 17 and 13 enzymes respectively. Despite this overlap, the two subgraphs have significantly different frequency patterns. The enzymes that are common to these subgraphs are of class transferases, lyases, ligases, isomerases, and oxireductases. These are common functions in both eukaryotic and prokaryotic organisms. The two subgraphs also have 10 common interactions. The remaining interactions define the difference between the eukaryotes and prokaryotes and they cause the dramatic drop in frequency value across the two clades. For instance, UMP kinase (2.7.4.22) is specific to prokaryotes for the synthesis of pyrimidines. Eukaryotes carry out the same function using UMP/CMP kinase (2.7.4.14), which is an enzyme used for phosphorylation of both CMP and UMP.

B. Comparison with the state-of-the-art method

We compare SiS with MULE [11] on the global map of metabolic network datasets. We obtained the executable of MULE from the authors. MULE finds all maximal frequent subgraphs with respect to any given frequency threshold

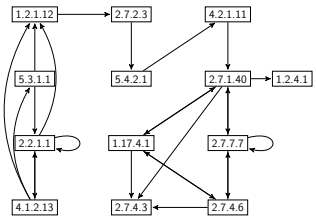

Figure 3. Maximal frequent size-25 subgraph in eukaryotes (global map metabolic pathways) discovered by both SiS and MULE.

value. This method is one of the closest methods to SiS in terms of the problem it considers. The difference is that it is seeking the maximal frequent subgraphs rather than a fixed size frequent subgraphs. To have a fair comparison, we extended our method to find maximal subgraphs as we described at the end of Section II-B.

Results on the Eukaryote-G dataset: We first experimented with Eukaryote-G dataset (145 organisms). We tested MULE with 83% frequency threshold and it found 925 frequent subgraphs in 20 minutes. The largest subgraph in the result set has 25 edges. We then lowered the frequency threshold to 80%. This increased the running time of MULE to 7 hour 27 minutes. It found three size-31 maximal frequent subgraphs. When we lowered the frequency threshold further to 70%, MULE did not finish in two days.

We then used SiS to find the top 50 most probable size-15 subgraphs in the same dataset. We extended these subgraphs to discover the maximal frequent subgraph. SiS generated the same size-25 maximal frequent subgraph as MULE (see Figure 3) in only 4.25 seconds. When we repeated this experiment with SiS to find the 50 most probable size-20, we found the same size-31 maximal frequent subgraph as MULE in only 4 minutes 18 seconds.

Results on the Prokaryote-G dataset: This dataset (1486 organisms) is much larger than the Eukaryote-G dataset (See Table I). MULE did not run on this dataset. Possibly this was because its memory requirements were too much. SiS however ran till completion in only 5.38 seconds for size-15 subgraphs and less than 5 minutes for size-20 subgraphs on the same dataset.

These results show that SiS can find the maximal frequent subgraph in a collection of graphs orders of magnitude faster than MULE. Also, it scales to large datasets for which MULE does not.

IV. Conclusion

In this paper, we developed a method, SiS, that efficiently discovers frequent subnetworks in a collection of biological networks. In other words, it looks for those subnetworks with the highest chance of appearing in a randomly drawn network from the input dataset. Our results demonstrate that our method scales to very large datasets (i.e., datasets with over a thousand networks, having a total of hundreds of thousands of nodes and edges) easily. For instance, it takes our method less than five minutes to find the maximal frequent subnetwork with up to 31 interactions in metabolic networks of 145 eukaryotes. It took MULE around 7.5 hours to find the same solution.

References

[1] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray, "From molecular to modular cell biology," *Nature*, vol. 402, pp. C47–C52, 1999.

[2] B. Palsson, *Systems Biology: Properties of Reconstructed Networks*. Cambridge University Press, 2006.

[3] Kaderali, Lars and Dazert, Eva and Zeuge, Ulf and Frese, Michael and Bartenschlager, Ralf, "Reconstructing signaling pathways from rnai data using probabilistic boolean threshold networks," *Bioinformatics*, vol. 25, no. 17, pp. 2229–2235, 2009.

[4] X. Yan and J. Han, "gSpan: Graph-Based Substructure Pattern Mining," in *ICDM*, 2002, pp. 721–724.

[5] S. Wernicke, "A faster algorithm for detecting network motifs," in *LNBI 3692*. Springer-Verlag, 2005, pp. 165–177.

[6] S. A. Cook, "The complexity of theorem-proving procedures," in *STOC*, 1971, pp. 151–158.

[7] S. Pemmaraju and S. Skiena, *Computational Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*. Cambridge University Press, 2003.

[8] J. Huan, W. Wang, and J. Prins, "Efficient Mining of Frequent Subgraphs in the Presence of Isomorphism," in *ICDM*, 2003, pp. 549–552.

[9] D. J. Cook and L. B. Holder, "Substructure discovery using minimum description length and background knowledge," *J. Artif. Int. Res.*, vol. 1, no. 1, pp. 231–255, Feb. 1994.

[10] M. Kuramochi and G. Karypis, "GREW-A Scalable Frequent Subgraph Discovery Algorithm," in *ICDM*, 2004, pp. 439–442.

[11] M. Koyutürk, A. Grama, and W. Szpankowski, "An efficient algorithm for detecting frequent subgraphs in biological networks," *Bioinformatics*, vol. 20, pp. 200–207, january 2004.

[12] M. Kanehisa and S. Goto, "KEGG: Kyoto Encyclopedia of Genes and Genomes," *Nucleic Acids Research*, vol. 28, no. 1, pp. 27–30, 2000.

[13] K. Dolinski and D. Botstein, "Orthology and Functional Conservation in Eukaryotes," *Annual Review of Genetics*, vol. 41, pp. 465–507, December 2007.

[14] N. A. Campbell and B. R. Jane, *Biology*. Benjamin Cummings, 2007.

[15] D. J. Richardson, B. C. Berks, D. A. Russell, S. Spiro, and C. J. Taylor, "Functional, biochemical and genetic diversity of prokaryotic nitrate reductases," *Cell Mol. Life. Sci*, vol. 58, no. 2, pp. 165–178, 2001.

[16] F. Keppler, J. T. G. Hamilton, M. Bra, and T. Röckmann, "Methane emissions from terrestrial plants under aerobic conditions," *Nature*, vol. 439, pp. 187–191, 2006.