BIOINFORMATICS

Vol. 00 no. 00 2004
Pages 1–8

An Efficient Algorithm for Detecting Frequent Subgraphs in Biological Networks

Mehmet Koyutürk*, Ananth Grama and Wojciech Szpankowski

Dept. of Computer Sciences, Purdue University, West Lafayette, IN 47907, USA.

ABSTRACT

Motivation: With rapidly increasing amount of network and interaction data in molecular biology, the problem of effectively analyzing this data is an important one. Graph theoretic formalisms, commonly used for these analysis tasks, often lead to computationally hard problems due to their relation to subgraph isomorphism.

Results: This paper presents an innovative new algorithm for detecting frequently occurring patterns and modules in biological networks. Using an innovative graph simplification technique, which is ideally suited to biological networks, our algorithm renders these problems computationally tractable. Indeed, we show experimentally that our algorithm can extract frequently occurring patterns in metabolic pathways extracted from the KEGG database within seconds. The proposed model and algorithm are applicable to a variety of biological networks either directly or with minor modifications.

Availability: Implementation of the proposed algorithms in the C programming language is available as open source at http://www.cs.purdue.edu/homes/koyuturk/pathway/.

Contact: Mehmet Koyutürk, koyuturk@cs.purdue.edu.

INTRODUCTION

Increasing availability of experimental data relating to biological sequences coupled with efficient tools such as BLAST and CLUSTAL have contributed to fundamental understanding of a variety of biological processes (Altschul *et al*., 1997; Thompson *et al*., 1994). These tools help in understanding relationships as well as differences between sequences and associated organisms. They are used for discovering common subsequences and motifs, which can be used to derive functional, structural, and evolutionary information. Recent developments in molecular biology have resulted in a new generation of experimental data that entails the relationships and interactions between biomolecules (Hartwell *et al*., 1999; Oltvai and Barabási, 2002). Biomolecular interaction data, generally referred to as biological or cellular networks, are frequently abstracted using graph models. Although vast amounts of high-quality data is becoming available, efficient analysis counterparts to BLAST and CLUSTAL are not readily available for such abstractions.

It is possible to model biological networks using various graph theoretic formalisms. Metabolic pathways, for instance, are naturally modeled using directed hypergraphs, with nodes representing compounds (substrates and products), and hyperedges representing enzymes (reactions). It is possible to reduce such a model into a general directed graph with nodes representing enzymes, and a directed edge from an enzyme to another implying that the product of the first enzyme is consumed by a reaction catalyzed by the other. Depending on the biological interpretation, if we are primarily interested in the existence of a producer-consumer relationship, we may also omit the direction from the edges.

As is the case with sequences, two key problems on graphs are: aligning multiple graphs, and finding frequently occurring subgraphs in a collection of graphs. Analysis of biological networks in terms of these problems provides understanding of several interesting concepts such as common motifs of cellular interactions, evolutionary relationships and differences among cellular network structures of different organisms, organization of functional modules, relationships and interactions between sequences, and patterns of gene regulation.

In this paper, we address the problem of finding frequently occurring subgraphs in a collection of metabolic pathways. This problem is particularly challenging because it relates to the NP-hard subgraph isomorphism problem. Therefore, appropriate modeling of biological networks is necessary in order to simplify the problem. We rely on a directed graph model with unique node labelings, which simplifies the graph mining problem significantly while being able to capture the underlying biological information accurately. We then devise an efficient algorithm that is based on frequent itemset extraction to discover frequent subgraphs among these graphs.

Experimental results on metabolic pathways extracted from the KEGG database demonstrate that our algorithm is able to discover interesting patterns (to a user-specified threshold on frequency) very quickly (within seconds). Furthermore,

*To whom correspondence should be addressed.

it provides a framework for multi-level analysis of enzymatic interactions by adjustment of support (frequency) thresholds. The proposed model and algorithm are also applicable to other biological networks either directly or with minor modifications.

In the next section, we discuss the use of graph theoretic formalisms for biological networks. We then present the proposed model for metabolic pathways and our algorithm for analyzing these pathways. In the Discussion section, we present and evaluate the experimental results obtained by running the proposed algorithm on KEGG metabolic pathway database. Finally, we draw conclusions and outline avenues for future research.

MODEL

Graph models are commonly encountered in computational analysis of cellular interactions (Olken, 2003). In the multilayered organization of organisms, such interactions form the bridge between individual molecules (*e.g.*, genes, mRNA, proteins and metabolites) and large-scale organization of the cell through functional modules (Oltvai and Barabási, 2002). Biological networks that represent cellular interactions can be in the form of metabolic pathways, signal transduction pathways, gene regulatory networks, and protein interaction networks. Efforts aimed at finding appropriate models for such networks have been motivated by significant advances in the understanding of genomics and have been the focus of considerable research attention.

Protein interaction networks are comprised of groups of interacting proteins, observed experimentally. They provide the experimental basis for the understanding of modular organization of the cells as well as useful information for predicting the biological function of individual proteins. Common methods of obtaining protein interaction data include two-hybrid experiments, mass spectrometry experiments, and structural analysis such as phage-display (Ho *et al*., 2002; Ito *et al*., 2001). Recently, there have been several efforts to organize protein interaction networks into publicly available databases such as BIND[1] and DIP[2]. The experimental data may reveal either, pairwise interactions, as in two-hybrid experiments, or multi-way interactions between a set of proteins, as in mass spectrometry experiments. Pairwise interactions are conveniently modeled by simple undirected graphs in which nodes represent proteins and an edge between two nodes represents the interaction between the corresponding proteins. Multi-way interactions are modeled using hypergraphs, in which edges are replaced by hyperedges that represent interactions between more than two proteins (Olken, 2003).

Gene regulatory networks, also referred to as genetic networks, represent regulatory interactions between pairs of genes and are generally inferred from gene expression data through microarray experiments (Akutsu *et al*., 1998). A simple and common mathematical model for gene regulatory networks is a Boolean network model. In this model, nodes correspond to genes and a directed edge from one gene to the other represents the regulatory effect of the first gene on the second. The edge is labeled by either a + or - sign to represent the direction of regulation, namely up- or down-regulation, respectively. More sophisticated computational models that capture the degree of regulation through weighted graphs and/or differential equations have also been proposed.

Metabolic pathways have a relatively longer history compared to other biological networks. They characterize the process of chemical reactions that, together, perform a particular metabolic function. With the recent progress in application of computational methods to cell biology, there have been successful attempts at modeling, synthesizing and organizing metabolic pathways into public databases such as KEGG[3] (Karp and Mavrovouniotis, 1994; Goto *et al*., 1997; Krishnamurthy *et al*., 2003). Metabolic pathways are chains of reactions linked to each other by chemical compounds (metabolites) through product-substrate relationships. A natural mathematical model for metabolic pathways is a directed hypergraph in which each node corresponds to a compound, and each hyperedge corresponds to a reaction (or equivalently, an enzyme) (Krishnamurthy *et al*., 2003). The direction of a pin of a hyperedge indicates whether the compound is a substrate or product of the reaction. It is possible to replace this model by a more simple directed graph if, for instance, we are only interested in relationships between enzymes. In such a model, enzymes correspond to nodes of the graph and a directed edge from one enzyme to another indicates that a product of the first enzyme is a substrate of the second. Indeed, metabolic pathways are represented in terms of various binary relations in KEGG (Goto *et al*., 1997). Edges may also be labeled by the compound that relates the two corresponding enzymes. An interesting property of metabolic pathways is the structure of these networks, which reflects temporal information. Specifically, an enzyme may show up more than once in the same pathway, implying that this enzyme takes part in the whole process at different time instants. The implication of this fact on graph models is that more than one node of a graph (pathway) might have the same label (enzyme). One might either be interested in preserving these temporal relations or only in general relations between pairs of enzymes. In the latter scenario, one may merge nodes in the graph with identical labels. We note that merging nodes with identical labels simplifies our graph analysis problem substantially. Furthermore, it is always possible to recover the existing temporal patterns from the generic patterns extracted from graphs with merged nodes.

[1] http://www.blueprint.org/bind/

[2] http://dip.doe-mbi.ucla.edu/

[3] http://www.kegg.com/

Graph-theoretic modeling of biological networks provides a framework for the solution of various problems aimed at understanding of molecular interactions in the cell. These problems include clustering, shortest-path computation, graph matching, graph alignment, subgraph homeomorphism, and graph mining (Olken, 2003). Clustering, graph alignment, and graph mining provide a suitable framework for identification of functional modules, which can be defined as a substructure of a biological network that is separable from other modules in terms of functionality. One approach to the identification of functional modules is graph clustering, or the discovery of dense subgraphs based on the assumption that a group of functionally related entities are likely to densely interact with each other while being somewhat separated from the rest of the network (Rives and Galitski, 2003). Another approach is multiple alignment of graphs or mining frequent subgraphs in order to discover common substructures in the network. The basis for this is that functional modules can be expected to repeat among several pathways and/or organisms (Tohsato *et al*., 2000). Graph alignment and graph mining also provide other opportunities for analyzing biological networks. The main focus of our study is on mining biological networks for frequent connected subgraphs.

APPROACH

Graph mining is a powerful tool for finding motifs and commonly occurring patterns in datasets that contain interactions. With the progression of molecular biology from sequences to biological networks, motif and pattern discovery becomes interesting and useful for such networks as for sequences. In this study, we devise algorithms for mining pathway substructures that are observed frequently over different organisms and/or metabolic pathways. While our focus here is on metabolic pathways, the models and algorithms can be applied either directly or with slight modifications to other biological networks as well.

Related Work on Graph Mining

Graph mining is a particularly challenging problem as it relates to the NP-hard subgraph isomorphism problem. This problem has attracted considerable interest recently since graph models appear in many scientific, commercial, and technological applications. Existing graph mining algorithms are generally based on frequent itemset mining, which is a well-studied problem in the data mining literature.

The definition and complexity of the graph mining problem varies significantly depending on the target application. For instance, a class of algorithms define the problem as one of finding isomorphic substructures (independent of labeling) in a collection of graphs, or equivalently, in a single large graph. This approach is well suited to applications focused on the structure of relationships between entities. However, it leads to a challenging computational problem since the hard subgraph isomorphism problem needs to be solved at every step of these algorithms. Consequently, research on this variation of the problem is mainly focused on efficient node and edge ordering schemes and related optimization techniques that help simplify the subgraph isomorphism problem (Kuramochi and Karypis, 2001; Han and Yan, 2002).

An alternate framework for graph mining defines the problem as one of finding frequent patterns that have both the entities (node labels) and relations between entities (graph structure) in common. This definition results in an easier problem and also suits the application of graph mining to biological networks, since we are mainly interested in common relations between biomolecules. One of the existing algorithms, Subdue, solves this problem with repeated enumerations, which can be computationally expensive for large-scale problems (Cook and Holder, 2000). Inokuchi *et al*. (2001) extend the a-priori algorithm for frequent itemset mining to this problem on an adjacency matrix model. This model also tends to be expensive for large sparse graphs. Our algorithm is based on frequent itemset mining as well. However, it takes advantage of the sparse nature of metabolic pathways to reduce the associated computational cost significantly.

Graph Formalism for Metabolic Pathways

We start our discussion by formally defining a metabolic pathway.

Definition 1. *A* **metabolic pathway** $P(M, Z, R)$ *is a collection of metabolites* M*, enzymes* Z*, and reactions* R*, where each reaction* $r \in R$ *is associated with a set of enzymes* $Z(r) \subseteq Z$*, a set of substrates* $S(r) \subseteq M$*, and a set of products* $T(r) \subseteq M$.

Our goal in mining metabolic pathways is to discover common motifs of enzyme interactions that are related to each other. Therefore, we model metabolic pathways with simple directed graphs that are able to capture the interaction information efficiently. Furthermore, we represent each enzyme by a unique node, independent of the number of times the enzyme appears in the underlying pathway. The purpose of this restriction is that it simplifies the graph mining problem significantly while providing results that are biologically meaningful. Moreover, this simplification does not cause any loss of information and the model can be easily reverted to capture more detailed information on pathways once frequent subgraphs are discovered.

Definition 2. *Given metabolic pathway* $P(M, Z, R)$*, the* **associated directed graph** $G(V, E)$ *of* P *is constructed as follows: for any enzyme* $z_i \in Z$*, there is a node* $v_i \in V$*. There is an edge from* v_i *to* v_j, i.e., $(v_i, v_j) \in E$ *if and only if* $\exists\, r_1, r_2 \in R$*, such that* $z_i \in Z(r_1)$*,* $z_j \in Z(r_2)$*, and* $T(r_1) \cap S(r_2) \neq \emptyset$.

This means that there exists a directed edge from one enzyme to another in the graph if and only if the second

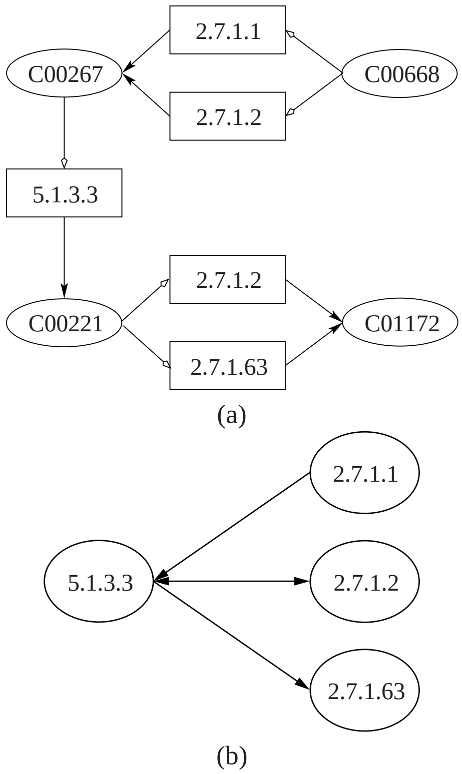

Fig. 1. Directed graph model for metabolic pathways: (a) a portion of glycolysis reference pathway, (b) its directed graph representation.

enzyme consumes a product of the first one. Figure 1 illustrates the directed graph model for metabolic pathways. In the pathway, enzymes are shown by rectangular boxes while metabolites are shown by ovals. Nodes, each corresponding to exactly one enzyme, are shown by ovals in the graph. Directions of the edges can be omitted to represent only interactions between enzymes without effecting the mining algorithm described in the next section.

An Efficient Algorithm for Mining Metabolic Pathways

The graph model described in the previous section simplifies the task of mining frequent subgraphs defined as follows:

Definition 3. *Given a collection of graphs $G_1, G_2, ..., G_n$ and support threshold ϵ, the* **Maximal Frequent Subgraph Discovery** *problem is one of finding all maximal connected subgraphs that are contained in at least ϵn of the input graphs.*

The above definition implicitly defines support, *i.e.*, the support of a subgraph that is contained by n' of the graphs is n'/n. A subgraph is frequent if its support is greater than ϵ. Connectivity ensures that the discovered frequent interactions are related to each other. Formally, a graph is connected if there is a path of subsequent edges between any two nodes in the graph. Maximality of discovered subgraphs is enforced in order to avoid redundancy. We say that a frequent subgraph maximal if it is not contained by another frequent subgraph, *i.e.*, its edge set is not a subset of edges of any other frequent subgraph.

Although graph mining is a hard problem in general, the above framework, which is well suited to our target application, simplifies the problem considerably. One reason for this is that subgraph isomorphism is no longer an issue as it is implicitly enforced by node labelings. Additionally, it is challenging to adapt existing data mining algorithms to graph mining since most of the existing data mining algorithms are based on problems with a single type of data unit (*e.g.*, item), while graphs contain nodes and edges as different types of data. In our model, uniqueness of nodes implies unique labeling of edges, providing us with the opportunity of reducing the problem to frequent itemset mining by specifying edges as fundamental data units. Since frequent itemset mining is extensively studied, and there exist many effective and well-tuned algorithms, we can adapt these algorithms to graph mining taking into account the nature of our problem.

Since a node label cannot be repeated in our directed graph model, every edge that may exist in a graph is uniquely specified by the labels of its incident nodes. This observation leads us to the idea of representing a connected subgraph by a set of edges, since the uniqueness of each edge implies uniqueness of a subgraph represented by a set of edges. We introduce the concept of a *connected edgeset* for this purpose to impose connectivity, since we are only interested in connected subgraphs by our problem definition.

Definition 4. *A* **unique edge** e *is a set of two node labels v_i, v_j. A set of unique edges $ES = \{e_1, e_2, ..., e_k\}$ is called a* **connected edgeset** *if and only if all edges in the set are connected,* i.e., *any subset $ES' \subset ES$ shares at least one node with the remaining set of edges $ES \setminus ES'$.*

We can now establish the link between maximal frequent connected subgraph discovery problem and frequent itemset mining problem, where graphs (pathways) correspond to transactions and connected edgesets correspond to itemsets. In frequent itemset mining, transactions are sets of items and the problem is one of finding all frequent itemsets that exist in more than a specified number of transactions. The fundamental approach used by frequent itemset mining algorithms is to construct frequent itemsets from smaller to larger sets based on the fact that any subset of a frequent itemset must be frequent. This is also true for edgesets in our problem. Consequently, enumerating all itemsets in a bottom-up fashion provides efficient pruning of the search space, since most large sets are eliminated without consideration.

We adapt the basic idea of frequent itemset mining to frequent subgraph mining with one additional constraint. Since we are interested in connected subgraphs, it is more efficient to consider only connected edgesets throughout the search process. While maintaining connectivity, it is also necessary to avoid redundancy, in terms of considering the same set of edges more than once in a different order. In order to handle these two issues efficiently, we develop a depth-first enumeration algorithm based on backtracking (Gouda and Zaki, 2001), which extends each subgraph with only edges from a candidate edge set. We maintain connectivity by only adding edges that are connected to the current subgraph and avoid redundancy by keeping track of already visited edges.

Another reason for selecting depth-first enumeration is that the major limitation in our application is memory size (as network databases become large). Since enumeration algorithms tend to require considerable memory, the time-space tradeoff in the design of algorithms may be resolved in favor of memory for this reason. Note that this does not imply that we compromise completely on runtime efficiency. Indeed, we show that our algorithm computes desired results within a few seconds for conventional databases. Note that adaptation of breadth-first data mining algorithms such as a-priori (Agrawal and Srikant, 1994) might be faster provided that there is sufficient memory.

The algorithm for frequent subgraph mining is shown in Figure 2. Upon each invocation, the algorithm tries to extend the edgeset (subgraph) by all edges in the candidate set one by one. If the extended edgeset is frequent, then the procedure is again invoked for the extended edgeset. The algorithm stops whenever an edgeset cannot be extended further. This edgeset is then recorded, if it is not contained by any other recorded frequent edgeset. In the algorithm, $N(e_i)$ denotes the neighbors of edge e_i, *i.e.*, it is the set of frequent edges that share at least one node with e_i. D, on the other hand, is the set of edges that are already visited by the algorithm. Therefore, while extending an edgeset, the neighbors of the newly added edge are added to the candidate set, while keeping the already visited edges out. The procedure is invoked as MinePathways(MFS, $\{e_i\}$, $N(e_i)$, $\{e_1, e_2, ...e_{i-1}\}$), for each edge e_i that is frequent in the collection of graphs. MFS is empty at first invocation, and it is input to the procedure at each subsequent invocation, by which it is extended with newly discovered frequent subgraphs.

Example. Consider the input graph collection of Figure 3(a). This collection has five edges in all, ab, ac, bd, ce, and de. Figure 3(b) shows the enumeration tree for mining subgraphs that exist in at least 3 of the input graphs. Procedure MinePathways is invoked for ab, ac and de, since these are the only frequent edges. Edges bd and ce are not considered since they are not contained in at least 3 graphs. The frequency of each edgeset is shown in parentheses. At the first invocation, the algorithm starts with edgeset $\{ab\}$, whose candidate set is $N(ab) = \{ac\}$ and extends it with edge ac as the edgeset $\{ab, ac\}$ is frequent. Since no further extension is possible, this edgeset is recorded as a maximal frequent subgraph. Note that extension of the edgeset with edge de is not considered since this edge is not connected to the edgeset under consideration, so it never gets into the candidate edge set. Furthermore, extension of the edgeset $\{ac\}$ with edge ab is not considered since this edge has already been visited. In the end, the algorithm reports two maximal frequent subgraphs, $\{ab, ac\}$ and $\{de\}$. Although edgeset $\{ac\}$ is also a leaf of the tree and is frequent, it is not reported since it is contained in another frequent edgeset.

procedure MinePathways(MFS, E_k, C_k, D)
  ▷ MFS: Set of maximal frequent subgraphs
  ▷ E_k: Frequent subgraph with k edges
  ▷ C_k: Set of candidate edges
  ▷ D: Set of already visited edges
  $ismaximal \leftarrow$ **true**
  for all edges $e_i \in C_k$ **do**
    $D \leftarrow D \cup \{e_i\}$
    $E_{k+1} \leftarrow E_k \cup \{e_i\}$
    if E_{k+1} is frequent **then**
      $ismaximal \leftarrow$ **false**
      $C_{k+1} \leftarrow (C_k \cup N(e_i)) \setminus D$
      MinePathways(MFS, E_{k+1}, C_{k+1}, D)
  if $ismaximal$ **then**
    if E_k has no superset in MFS **then**
      $MFS \leftarrow MFS \cup E_k$

Fig. 2. Depth-first enumeration algorithm for mining maximal frequent subgraphs.

DISCUSSION

Using the proposed algorithm, we mine several pathway collections extracted from the KEGG metabolic pathway database. KEGG currently has a fairly comprehensive database of metabolic pathways. KEGG also has a base of reference pathways that can be viewed as networks of enzymes, which are constructed manually. Organism-specific pathways are then constructed automatically with the help of identified enzyme genes. By the end of 2003, KEGG contained pathway maps of several metabolic processes, including carbohydrate, energy, lipid, nucleotide, and aminoacid metabolism for 157 organisms.

We mine several pathways belonging to different metabolisms for different organisms. Sample frequent sub-pathways discovered in pathway collections that belong to glutamate, alanine-aspartate, and pyrimidine metabolisms are shown in Figures 4, 5 and 6, respectively. The nodes of the displayed graphs are labeled by KEGG ID's of enzymes, which can be queried on KEGG web site for detailed information.

We are able to observe considerably large sub-pathways that are frequent. For example, a sub-pathway of glutamate metabolism that contains 4 nodes and 6 edges occurs in 45 (29%) of the 155 organisms. This sub-pathway is shown

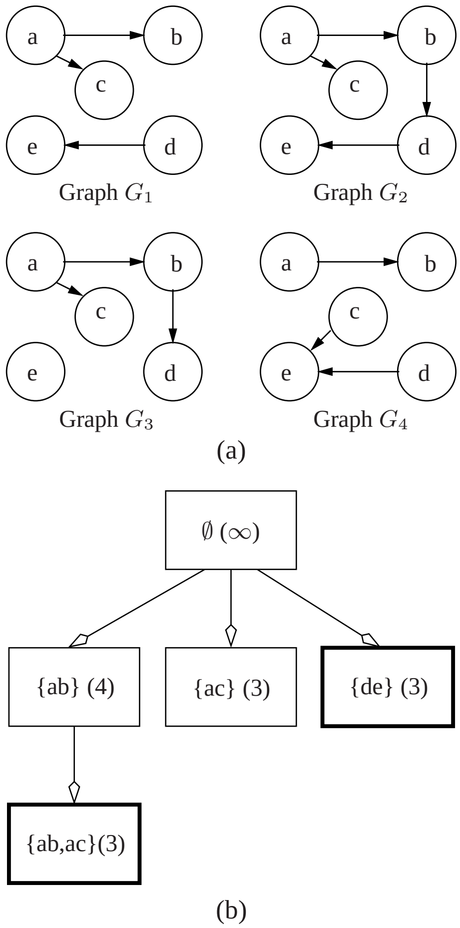

Fig. 3. Sample execution of frequent subgraph mining. (a) Input collection of graphs, (b) resulting enumeration tree of frequent edgesets.

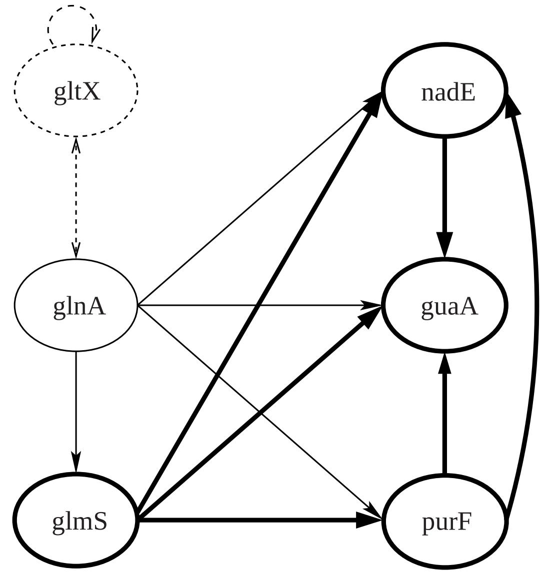

Fig. 4. Frequent sub-pathways discovered for different support values on glutamate metabolism among 155 organisms.

by bold nodes and edges in Figure 4. It is composed of enzymes glmS (2.6.1.6 - glucosamine-fructose-6-phosphate-aminotransferase), guaA (6.3.5.2 - GMP synthase), nadE (6.3.5.1 - NH(3)-dependent NAD(+) syhthetase), and purF (amidophosphoribosyltranferase). In this sub-pathway, all enzymes are related by L-Glutamine.

Mining the pathways for different support thresholds allows evaluation of frequent sub-pathways in a multi-level fashion. For instance, when we reduce the required support threshold to 19.3% (30 organisms) for glutamate metabolism, the largest sub-pathway we are able to discover consists of 5 nodes and 10 edges, which is indeed a supergraph of the previous one. This sub-pathway is shown in the figure by solid nodes and edges. As seen in the figure, this pathway contains enzyme glnA (6.3.1.2 - glutamine synthetase), which is also related to the other enzymes by L-glutamine. Further reducing the support threshold to 14.2% (22 organisms), we are able to discover a sub-pathway of 6 nodes and 13 edges, which is the entire graph shown in the figure. This pathway is indeed a supergraph of the previous one, with gltX (6.1.1.17 - glutamyl-tRNA synthetase) added, which interacts bidirectionally with glnA through L-Glutamate. The self-loop for gltX implies that this enzyme takes part in two consecutive reactions, which are part of the observed frequent sub-pathways.

In Figure 5, largest of the frequent sub-pathways that are discovered in alanine-aspartate metabolism for three different levels of support threshold are shown. The bold sub-pathway of 5 nodes and 8 edges occurs in 50 (32.1%) of the 156 organisms, the solid one with 5 nodes and 11 edges occurs in 30 (19.2%) of the organisms, and the entire graph of 6 nodes and 16 edges occurs in 18 (11.5%) of the organisms. Note that enzyme purB (4.3.2.2 - adenylosuccinate lyase) and its interaction with purA (6.3.4.4 - adenylosuccinate synthetase) through adenylosuccinate (N6-(1,2-Dicarboxyethyl)-AMP), shown in dotted lines in the figure, is included in the most frequent sub-pathway of alanine-aspartate metabolism but is excluded from the larger sub-pathways of lower frequency, which is interesting to note.

Figure 6 shows the multi-level analysis of frequent sub-pathways for pyrimidine metabolism. The bold sub-pathway of 4 nodes and 5 edges occurs in 40 (25.6%) of the 156 organisms, the solid one with 5 nodes and 7 edges occurs in 34 (21.8%) of the organisms, and the entire graph of 7 nodes and 11 edges occurs in 24 (15.4%) of the organisms.

Table 1 shows the results obtained from mining different metabolic pathway collections for varying level of minimum support. In this table, we report the number of discovered maximal frequent sub-pathways, number of edges in the largest discovered sub-pathway and the running time in seconds for the three metabolisms in discussion. Glutamate pathway

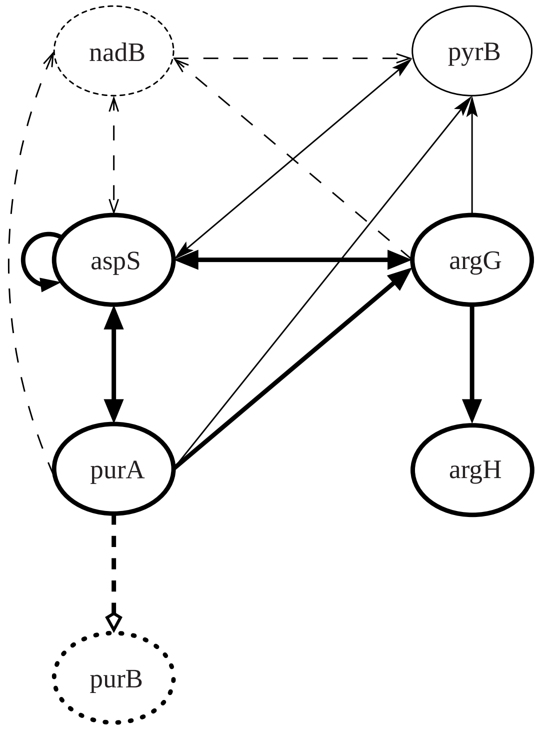

Fig. 5. Frequent sub-pathways discovered for different support values on alanine-aspartate metabolism among 157 organisms.

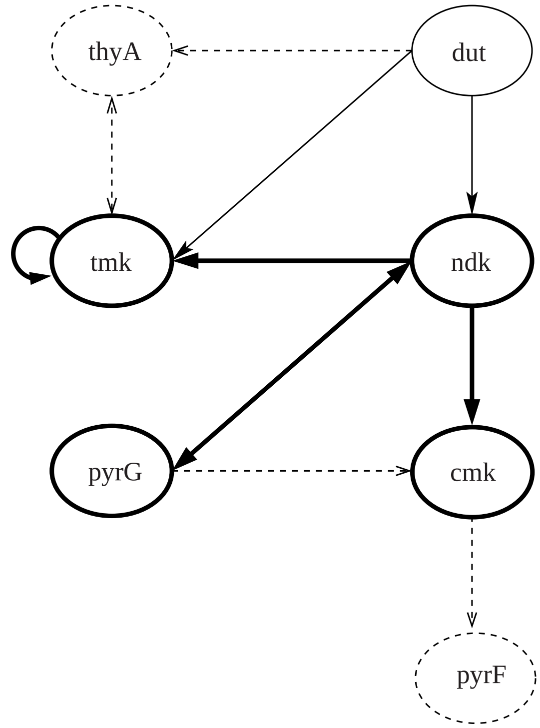

Fig. 6. Frequent sub-pathways discovered for different support values on pyrimidine metabolism among 156 organisms.

collection has a total of 2804 nodes and 11339 edges over 155 organisms, alanine-aspartate pathway collection has 2681 nodes and 8481 edges over 156 organisms, and pyrimidine pathway collection consists of 3375 nodes and 7218 edges over 156 organisms. On a Pentium-IV 2.0 GHz workstation with 512 MB RAM, we are able to mine these pathway collections in less than a second for relatively high support values to obtain meaningful results in terms of the size of the discovered frequent sub-pathways. For lower values of support, many sub-pathways turn out to be frequent and the size of the frequent pathways also grows significantly. For this reason, it takes more time for the algorithm to return all frequent sub-pathways. This is still extremely fast since the number of possible sub-pathways grows exponentially with the size of the sub-pathway. We are able to discover a sub-pathway of 16 edges, which is considerably large, in only three seconds. Therefore, we conclude that the proposed algorithm provides near real-time response for practically interesting queries in much the same way as state-of-the-art sequence matching algorithms such as BLAST.

The implementation of the proposed mining algorithms in the C programming language is available as open source at http://www.cs.purdue.edu/homes/koyuturk/pathway/. Some sample results for multi-level analysis of frequent sub-pathways are also provided at this web site.

Table 1. Time spent on mining different metabolic pathways for varying level of minimum support.

Metabolism	Min. Supp. (%)	# Frequent Subpathways	Largest # Edges	Runtime (secs.)
Glutamate	10.0	34	15	0.52
	12.5	39	13	0.17
	15.0	21	11	0.03
	20.0	12	9	0.00
Pyrimidine	10.0	120	15	0.44
	12.5	78	15	0.19
	15.0	49	12	0.04
	20.0	23	7	0.00
Alanine and Aspartate	10.0	34	16	3.08
	12.5	25	16	1.84
	15.0	21	12	0.15
	20.0	15	11	0.02

CONCLUSION

With the rapidly increasing amount of network and interaction data in molecular biology, the problem of mining patterns, motifs, and modules in biological networks becomes increasingly interesting. This paper provides a framework for mining biological networks using an innovative graph simplification, which leads to efficient graph mining algorithms. The proposed model and algorithm are shown to be well-suited to mining metabolic pathways providing interesting results and being able to respond to queries quickly. It also provides a framework for multi-level analysis of occurrence of sub-pathways in metabolic pathways. Our approach can be easily extended to other biological networks as well.

The proposed framework can be further improved by adding flexibility for capturing biologically meaningful information that helps in interpretation of discovered patterns. An important improvement in this respect is the investigation of possible probabilistic models and metrics to help the evaluation of statistical significance of the discovered patterns. Finally, the concept of a matching subgraph can be extended to one of an "approximate match". The notions of approximations and distance would need to be formalized before such algorithms can be devised.

ACKNOWLEDGEMENTS

This work was supported by NSF Grants CCR-9804760 and CCR-0208709, and NIH grant R01 GM068959-01.

REFERENCES

Agrawal,R. and Srikant,R. (1994) Fast algorithms for mining association rules, *Proc. 20th Intl. Conf. Very Large Data Bases (VLDB'94)*, 487-499.

Akutsu,T., Kuhara,S., Maruyama,O. and Miyano,S. (1998) Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions, *Proc. Ninth Annual ACM-SIAM Symp. Discrete Algorithms*, 695-702.

Altschul,S.F., Madden,T.L., Scheffer,A.A., Zhang.Z, Zhang.J, Miller,W. and Lipman, D.J. (1997) Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research*, **25**, 3389-3402.

Cook,D.J. and Holder,L.B. (2000) Graph-based data mining, *IEEE Intelligent Systems*, **15**, 32-41.

Goto,S., Bono,H., Ogata,H., Fujibuchi,W., Sato,K. and Kanehisa,M. (1997) Organizing and computing metabolic pathway data in terms of binary relations, *Pacific Symp. Biocomputing*, 175-186.

Gouda,K. and Zaki,M.J. (2001) Efficiently mining maximal frequent itemsets, *IEEE Intl. Conf. Data Mining (ICDM'01)*, 163-170.

Hartwell,L.H., Hopfield,J.J., Leibler,S. and Murray,A.W. (1999) From molecular to modular cell biology, *Nature*, **402**, C47-C51.

Ho.Y, *et al*. (2002) Systematic identification of protein complexes in Saccharomyces cerevisae by mass spectrometry, *Nature*, **415**, 180-183.

Inokuchi,A., Washio,T., Okada,T. and Motoda,H. (2001) Applying the apriori-based graph mining method to mutagenesis data analysis, *Journal of Computer Aided Chemistry*, **2**, 87-92.

Ito,T., Chiba,T., Ozawa,R., Yoshida,M., Hattori,M. and Sakaki,Y. (2001) A comprehensive two-hybrid analysis to explore the yeast protein interactome, *PNAS*, **98**, 4569-4574.

Karp,P.D. and Mavrovouniotis,M.L. (1994) Representing, analysing, and synthesizing biochemical pathways, *IEEE Expert*, 11-21.

Krishnamurthy,L., Nadeau,J., Özsoyoğlu,G., Özsoyoğlu,M., Schaeffer,G., Taşan,M. and Xu,W. (2003) Pathways database system: An integrated system for biological pathways, *Bioinformatics*, **19**, 930-937.

Kuramochi,M. and Karypis,G. (2001) Frequent subgraph discovery, *IEEE Intl. Conf. Data Mining (ICDM'01)*, 313-320.

Olken,F. (2003) Biopathways and protein interaction databases, A lecture in *Bioinformatics Tools for Comparative Genomics: A short course*.

Oltvai,Z.N. and Barabási,A.L. Life's complexity pyramid, *Science*, **298**, 763-764.

Rives,A.W. and Galitski,T. Modular organization of cellular networks, *PNAS*, **100**, 1128-1133.

Thompson,J.D., Higgins,D.G. and Gibson,T.J. CLUSTAL-W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice, *Nucleic Acids Research*, **22**, 4673-4680.

Tohsato,Y., Matsuda,H., and Hashimoto,A. (2000) A multiple alignment algorithm for metabolic pathway analysis using enzyme hierarchy, *Eighth Intl. Conf. Intelligent Systems for Molecular Biology (ISMB'00)*, 376–383.

Yan,X. and Han,J. (2002) gSpan: Graph-based substructure pattern mining, *IEEE Intl. Conf. Data Mining (ICDM'02)*, 721-724.