Mining Edge-disjoint Patterns in Graph-relational Data *

Christopher Besemann and Anne Denton
North Dakota State University
Department of Computer Science
Fargo ND, USA
christopher.besemann@ndsu.edu, anne.denton@ndsu.edu

Abstract

Diverse types of data are associated with proteins, including network and categorical data. While graph mining techniques have long focused on data with no more than one label per node, generalizations have recently been developed. We show that existing generalizations are not well suited to typical biological networks and are likely to return few or no results on protein regulatory networks. They are, furthermore, ill-suited to graphs that are dense or show the small world property, which are typical features of biological networks. A graph-relational edge disjoint instance mining algorithm (GR-EDI) is presented that resolves these problems. Our algorithm treats bipartite edges separately and only constrains unipartite edges to be disjoint. We introduce a new pattern constraint that recovers the downward closure property. The algorithm uses a search lattice traversal strategy that allows more effective mining of graphs that cannot be considered as sparse due to hubs. Effectiveness is demonstrated for a real biological example. While existing techniques return few or no patterns, GR-EDI is able to extract many patterns.

1 Introduction

High-throughput experimentation is producing a wealth of information related to proteins. Some data have graph characteristics, such as regulatory networks [23], physical [13] and genetic [24] protein-protein interaction networks, and domain fusion networks [19]. Network patterns are important at every level of the organization of a cell, and are studied extensively in systems biology [21]. Frequently occurring network patterns, or network motifs, have been identified as important building blocks [20] of many networks.

Protein information is not limited to networks but also includes functional and localization information, which is maintained by the Gene Ontology Consortium [5], as well as sequence information. Extending frequent pattern mining to such diverse data can be expected to lead to even richer network patterns. However, traditional graph-theory and many frequent pattern mining algorithms [16, 29] focus on graphs with no more than one label per node. Given the rich information associated with proteins, the limitation to a single node label is problematic.

In this paper, we consider protein networks in combination with sets of protein properties, or properties for short. An existing approach towards generalizing graph-theoretical concepts to multi-labeled graphs is to consider the relationship between each protein and each of its properties as a bipartite graph. The bipartite graph is then combined with the unipartite graph that characterizes the protein network in question [17]. A benefit of this approach is that it is straightforward and does not require many, if any, modifications to existing algorithms for single-labeled graphs. We show that the approach of treating multiple labels as part of a combined graph leads to a major problem: It might be expected that in the limiting case of a single property association per node, the traditional single-label approaches and the graph-relational generalization should lead to the same results. However, for edge-disjoint graph mining algorithms [17, 26] this is not the case.

Edge-disjoint instance (EDI) graph mining is an important branch of frequent subgraph mining, in which the support of a pattern is based on instances that do not share edges. This approach often produces more intuitive results than simple frequent pattern mining, as is discussed in [17]. It has the additional benefit that support measures used in EDI algorithms have the downward closure property that allows pruning of the search lattice [26]. Algorithms that allow arbitrary overlap in pattern instances do not have these properties. However, requiring edge disjointness for the bipartite graph that relates proteins with properties is problematic in biological networks. Parallel or alternate pathways, in which the same function of a protein contributes to multiple instances of the pattern are often of interest [18], and the edge disjointness criterion, hence, has to be considered as too strict. In fact, we show that expecting disjointness for bipartite edges that connect proteins with properties is not even in accordance with edge disjoint mining of single label graphs. If proteins were only allowed to have a single property as label, and single label EDI mining was applied, then the same protein with the same label could contribute to multiple instances of the same pattern. This is because the relationship between the protein and its label would not be considered an "edge" in the traditional single-label setting.

We introduce a graph-relational EDI algorithm (GR-

*Supported by the NDSU Presidential Fellowship program and the National Science Foundation under Grant No. IDM-0415190.

EDI) that recovers the results of traditional single-property graph mining within the graph-relational setting by excluding bipartite edges from the disjointness condition. We are able to maintain the downward closure property by defining a graph-relational pattern constraint. GR-EDI is shown to produce more biologically useful results for a practical bioinformatics problem compared to general EDI.

Existing EDI algorithms focus on sparse graphs. However, many biological networks are so dense, due to hubs, that researchers have had to limit the maximum degree of nodes by deleting nodes with a degree higher than a threshold [17]. Such approaches have drawbacks, especially since hubs are often of particular interest to biologists. Our algorithm uses a special lattice traversal order that is more effective than existing ones at exhaustively returning all patterns of a particular unipartite shape size. Results are not limited in the number of properties in the pattern, even for limited memory resources. This ordering of the search lattice traversal is particularly suitable to biological networks, since such networks often have the small-world property [27]. Patterns are expected to relate entities that are close. Hence, the size of the unipartite shape can safely be constrained to be smaller than the diameter of the unipartite graph.

2 Related Work

Much work has been done on understanding the structural properties of biological networks [20, 27]. Pattern mining algorithms have been developed for mining biological graphs, in particular graph transactions, using special pattern types [28, 11, 15]. Some work has also been done on mining network motifs within single input graphs [20, 14, 4]. NeMoFinder [4] stresses the importance of allowing overlap between patterns in biological networks. The algorithm itself does not use measures that are downward closed since the authors seek to find arbitrary overlap between all edges of motif patterns.

Related approaches can also be found in the general graph data mining literature. Data mining of graph transactions addresses the problem of finding subgraphs in a set of input graphs [29, 12, 16]. In this paper we address the problem of mining graph patterns from a single input graph, which is also discussed in [7, 25, 17]. Complexity is a more important concern in pattern mining on single input graphs because arbitrary overlap in pattern instances limits opportunities for pruning [17].

Edge-disjoint instance (EDI) mining is one approach towards pruning in the single input graph setting. Kuramochi and Karypis developed the concept of frequent edge-disjoint instances using the greedy-maximal independent sets (GMIS) algorithm [17]. Their algorithms HSIGRAM and VSIGRAM traverse the frequent subgraph lattice in a horizontal and vertical fashion respectively. This extends research on EDI initiated by Vanetik et al. in [25].

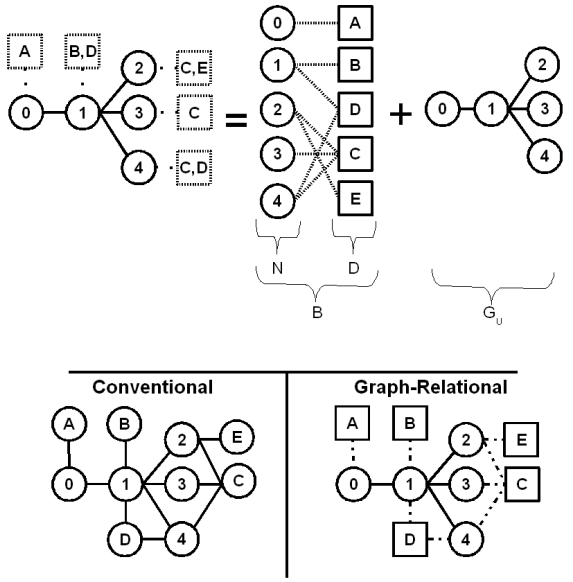

Figure 1: Biological networks viewed with set-labeled nodes (top left) are composed of a bipartite graph that includes attribute nodes (B, top center) and a unipartite graph of entities (U, top right). The single-labeled version (lower left) can be viewed in the graph-relational setting (lower right) where bipartite edges are not treated as true edges.

Vanetik et al. define conditions of admissible support measures on an overlap graph and give proofs in [26].

Few approaches have been developed that combine the analysis of protein network structure and protein properties [22, 3, 2]. None of these use the EDI concept. In the data mining literature, multi-relational mining addresses general questions that can include arbitrary relational data [6, 9]. These approaches are very general but they do not scale to the large and dense biological networks of interest in this paper. Kuramuchi et al. have also shown how graph mining techniques can be adapted to address set-based data associated with graph nodes [17]. Biological data is used as one of their examples.

Our GR-EDI algorithm differs from previous EDI work in two important aspects. In order to allow overlap of protein function we make a distinction between unipartite and bipartite relationships. We also introduce a more efficient search lattice traversal that is better suited to the dense nature of biological networks.

3 Preliminaries and Notation

We consider data characterized by two graphs: A unipartite graph of entity nodes and a bipartite graph that relates entity nodes and descriptor nodes. Entity nodes can, for example, represent genes or proteins. Descriptor nodes may stand for protein properties, such as functions. Without loss of

generality, entity-nodes are assumed to be unlabeled. Figure 1 illustrates the setting. The top left represents the biologist perspective of objects, such as proteins, with their set of properties. The top right shows how the same information can be represented as a bipartite and a unipartite graph. Conventional EDI approaches [17] combine both graphs into one, single-labeled graph (bottom left). We, in contrast, maintain a distinction between entity and descriptor nodes (bottom right) and refer to it as the *graph-relational* setting.

Formally, for the conventional setting, let $N = \{n_1, \ldots, n_n\}$ be a set of n "entity" nodes that have a structural (unipartite) relationship $U \subseteq (N \times N)$ between entities see Figure 1 (top right). Also let $D = \{d_1, ..., d_m\}$ be a set of m descriptor (property) nodes. D represents the domain of properties of proteins or other entities. The bipartite relation $B \subseteq (N \times D)$ is represented in Figure 1 top center. The conventional approach considers the graph $G_{gb}(V, E, L_v)$ where $V = \{N \cup D\}, E = \{U \cup B\}$. The set L_v denotes vertex labels. Edges in general may have labels, but for simplicity we assume that they are unlabeled. Each descriptor node is considered to have a unique label and each entity node to be unlabeled. In this single-labeled graph-based setting the nodes in V and edges in E are not distinguishable as in the lower left of Figure 1.

DEFINITION 3.1. A (subgraph) pattern $P(V, E, L_v)$ is a connected graph composed of any number of entity nodes, descriptor nodes, and edges.

To denote a subgraph pattern, we designate numbered vertices as entity nodes and lettered vertices as the labels of descriptor nodes. For example, $\{(0,1),(0,a)(1,b)\}$ denotes a subgraph of two nodes with one arc from one to the other where the source is linked to descriptor with label "a" and the target is linked to descriptor with label "b". Note that numbers in patterns do not correspond to node ids in the database.

G_s is a subgraph of G_{gr} if and only if $V_s \subseteq V$ and $E_s \subseteq E$. Two graphs G_1 and G_2 are isomorphic if there is a bijection $\rho : V_1 \rightarrow V_2$ where $l(v_i \in V_1) = l(\rho(v_i) \in V_2)$ such that $(v_i, v_j) \in E_1 \Leftrightarrow (\rho.v_i, \rho.v_j) \in E_2$ (there is a bijection on edges that respects the vertex mapping and vertex labels). The number of isomorphic embeddings of G_s in G_{gr} is the number of distinct instances of G_s in G_{gr}. Two occurrences of a connected graph G_{s_1} and G_{s_2} are distinct when $G_{s_1}.E \neq G_{s_2}.E$. Connected graphs have a path between any two vertices. Given a frequency threshold τ, a pattern or subgraph G_s is frequent in G if $Supp(G_s) \geq \tau$ where $Supp(G_s)$ represents the frequency of the pattern in the database.

3.1 Edge-disjoint (EDI) approaches In the single graph setting the number of instances of a pattern does not have the downward closure (anti-monotone) property [17, 8, 26] of the famous Apriori algorithm in market basket research [1]. Methods that consider edge-disjoint instances of a pattern solve this problem by redefining support. Vanetik et al. describe the concept of EDI using instance (overlap) graphs [26] and prove that the maximal independent set (MIS) over the instance graph is an admissible support measure. An overlap graph is based on all instances of pattern P within an input graph G. Two instances of P, u and v that are subgraphs of G are said to overlap or intersect if their edge-sets overlap or $u.E \bigcap v.E \neq \emptyset$.

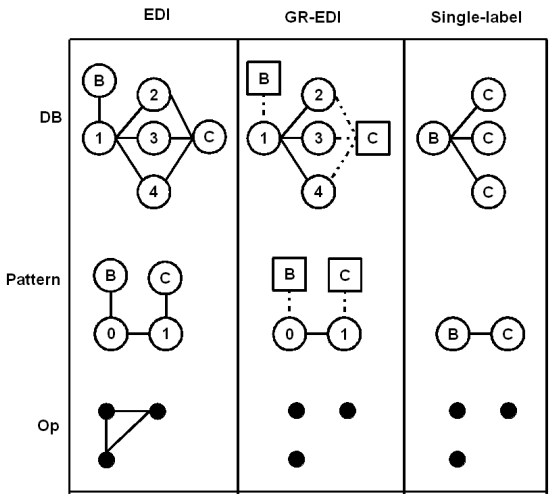

Figure 2: Overlap settings under a standard EDI algorithm (first), under the GR-EDI definition that ignores bipartite edges (second), and using a single-labeled graph with no bipartite edges where EDI and GR-EDI are equal.

The overlap graph $O_P = (I_p, E_I)$ is a graph that represents the distinct instances of P found in G_{gb} as a set of nodes I_P. E_I is the set of edges (u, v) that connect two instances if and only if the two instances overlap. Figure 2 left illustrates the overlap graph of an example graph that was constructed by combining a unipartite and a bipartite graph in the conventional way. Only one instance would be counted, since there is only one fully connected component in the overlap graph. The example graph happens to only have a single label per node, so we can also look at its representation as a single-labeled graph (right). It can be seen that the overlap graph in the right representation has no connecting edges, i.e. three instances would be counted. The center shows how the same example would be treated by our graph-relational GR-EDI algorithm, which we discuss next.

4 GR-EDI Patterns and Overlap

The GR-EDI algorithm considers the data as the graph formed by the union of the two edge types and node types that preserves the identity of the source graphs. The graph $G_{gr}(V, E, L_v, T_v, T_e)$ has a vertex (V) set, an edge (E) set,

and labels (L_v) as in the general case. We simplify by assuming unlabeled entity nodes and edges. A type label T_v is introduced to distinguish entity nodes and descriptor nodes. In our case, entity nodes are the unlabeled nodes (denoted as different node shape in Figure 1, bottom right). A type label T_e is also given to edges to distinguish unipartite and bipartite edges. In the following, we refer to vertices as $G_{gr}.N$ for entity nodes and $G_{gr}.D$ for descriptor nodes, and to unipartite edges as $G_{gr}.U$ and bipartite as $G_{gr}.B$.

DEFINITION 4.1. The graph-relational pattern constraint allows a pattern P if $P' = P/(P.D, P.B)$ is a connected graph. That is the pattern over unipartite edges and entity nodes only is connected.

Figure 2 illustrates the difference between GR-EDI overlap, standard EDI overlap, and single-labeled graph definitions. GR-EDI does not consider the pattern instances as overlapping since bipartite edges are not considered in the overlap calculation. Notice, how this approach reduces to the single-label overlap graph when entities have only one label and the graph is treated as a single-labeled graph. In the standard graph mining setting, any connected subgraph can be a pattern. We introduce a constraint into graph-relational pattern mining that only allows patterns, in which the entity nodes are connected through unipartite edges alone. Figure 3 shows the general pattern lattice where dotted boxes indicate patterns that fail the constraint. The GR-EDI patterns meet the constraint by connecting two entity nodes.

DEFINITION 4.2. A graph-relational overlap graph $O_P(I_P, E_P)$ is defined where two pattern instances $u, v \in I_P$ form the edge $(u, v) \in E_P$ if and only if their unipartite edges overlap ($u.U \bigcap v.U \neq \emptyset \Leftrightarrow (u, v) \in E_P$).

4.1 GR Downward Closure The downward closure property of the EDI algorithms [25, 17] is maintained by finding the MIS of the standard overlap graph. MIS applied to the GR-EDI overlap graph also maintains the downward closure property. Vanetik et al. identified that in order to be admissible a support measure based on the overlap graph must be non-decreasing under the operations of clique contraction, (connected) node addition, and edge removal (see [26] for full definitions and proof). Connected node addition means that a node is added and linked to *all* other nodes in the graph. These operations are required to transform the overlap graph of a pattern to the overlap graph of a sub-pattern. Therefore, to maintain downward closure the operations must be non-decreasing so that the support of the sub-pattern is at least as large as the support of the pattern. We follow the same reasoning with GR-EDI by showing that none of the required operations on the GR-EDI overlap graph result in increasing support.

In GR-EDI, the overlap graph changes when the underlying pattern is modified by either adding a unipartite or bipartite edge. Unipartite edge addition is similar to the original single-label graph changes. It results in the following potential changes for any instance in the overlap graph:

1. Instance has no valid edge for addition
2. Instances merge by sharing the added edge
3. Instance branches due to multiple edge addition

The existing overlap graph changes by node deletion, edge addition, or clique expansion for the cases. These changes correspond to the three operations required to move from pattern to sub-pattern overlap graphs. Bipartite edges make the following connections:

1. Existing entity node to an existing descriptor node
2. Existing entity node to an new descriptor node
3. New entity node to an existing descriptor node

In the first two cases, the corresponding instance either remains a node in the overlap graph or it is removed according to whether it is able to add the edge or not. The third case causes MIS to fail downward closure if no unipartite edges are present in the pattern. The original instance node can branch into multiple *independent* instance nodes when the bipartite edge is added to the new entity node. For example, a simple pattern of (0,b) contains one bipartite edge and zero unipartite edges. If this pattern was to be extended by adding another bipartite edge to form (0,b)(1,b) then instances of this pattern would never overlap by the GR-EDI definition. The new overlap graph would add disconnected nodes to the existing one see Figure 3. In GR-EDI entity nodes in patterns are constrained to be connected through unipartite edges, thereby avoiding this problem. Note also that the initial pattern (0,b) is not valid in GR-EDI; we must start with the unipartite pattern shape (0,1).

5 GR-EDI Algorithm

The GR-EDI algorithm can be viewed as a hybrid based on the HSIGRAM algorithm presented in [17]. We explore the lattice of the unipartite graph in a horizontal fashion that is analogous to the HSIGRAM algorithm. Bipartite edges are added apriori-like at each shape, thereby moving vertically as opposed to a pure horizontal approach. The motivation for this hybrid approach is that patterns that are small in their unipartite connectivity can be explored with a relatively small memory cost. For dense graphs, patterns with few unipartite edges can be returned exhaustively even if solving the full problem exceeds memory requirements. Two further changes were made with respect to the original EDI-type algorithm. First, bipartite edges are not considered in constructing the overlap graph. Second, the graph-relational

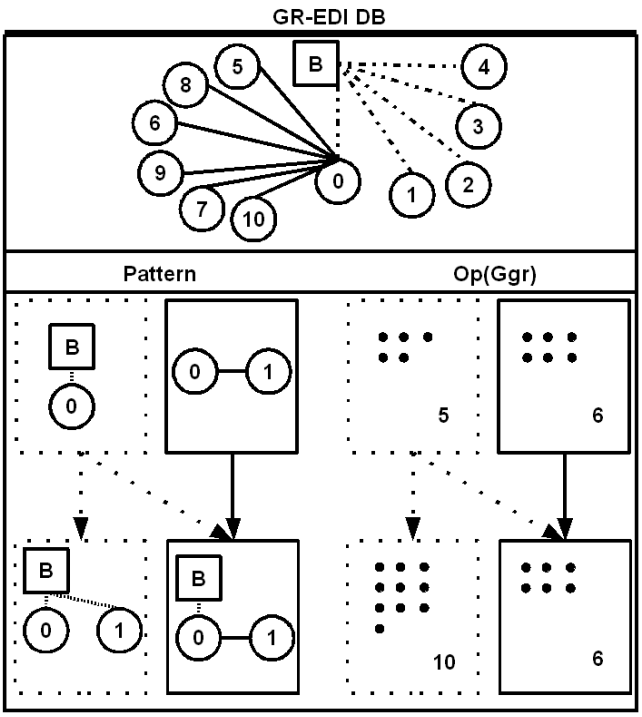

Figure 3: Example of a pattern lattice. Solid boxes are valid GR-EDI patterns. Solid lines indicate downward closure. Overlap graphs are shown for the sample database, notice that general patterns may not hold to downward closure.

constraint on patterns is enforced, which states that all entities are required to be connected through unipartite edges alone.

GR-EDI initializes with the frequent 1,* and 2,*-patterns (all patterns with one and two unipartite edges) since the main loop joins frequent k,0-patterns (unlabeled unipartite shapes) based on common (k-1),0 core patterns, see [17] for details on candidate pruning. This results in patterns that meet the GR-EDI constraint since no bipartite edges are added yet. If the instances are above the support threshold, then the EDI property is tested. The greedy MIS algorithm is applied as in [10, 17]. Finally, once a unipartite shape is identified as frequent, all bipartite extensions are discovered in an apriori-like algorithm. This algorithm works almost exactly like the outer-loop process but candidates are only allowed to add bipartite edges to maintain the GR-EDI pattern constraint. Downward closure is tested against the GR-EDI lattice.

MIS is computed over the overlap graph for each bipartite enhanced pattern. This step uses the GR-EDI definition of overlap compared to the single-label version that would have considered all edges in the overlap. The memory cost of each pattern search at this stage is no more that the cost from the previous main loop search for the unlabeled pattern

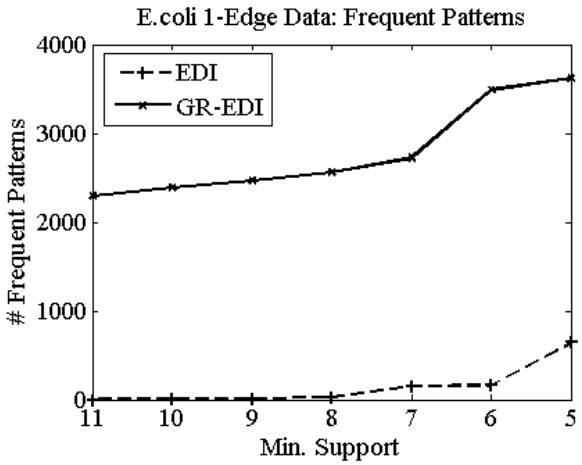

Figure 4: Frequent patterns in *E.coli* network.

as the intermediate calculations based on the overlap graph will always be smaller than the parent pattern.

6 Experiments and Discussion

We apply GR-EDI to a real *E.coli* data set, in which the unipartite entity relationship represents gene regulatory interactions. The bipartite relationship describes gene and protein properties such as domain annotations. There are a total of 3,241 directed edges between 1,362 nodes, and 2,841 descriptors that form 6,674 bipartite edges with the nodes. The data are described in detail in [2]. We first compare GR-EDI algorithm and the EDI algorithm based on the patterns that each approach is able to produce. We see a significant difference in the number of patterns recognized for each technique, see Figure 4. Notice that at the initial support setting, EDI was unable to produce any results. As support is lowered, EDI finds some patterns but GR-EDI continues to return substantially more patterns.

The large difference is primarily due to hub genes that are linked to a large number of other genes. In other words, the general EDI approach only allows proteins to contribute the same function to one disjoint regulatory "pathway" while GR-EDI allows each protein to contribute to as many disjoint pathways that exist in the network. This property of GR-EDI is especially important when the biological network has a structure centered on a smaller number of originating nodes such as regulators. In the regulatory network there are only few regulators (143 for our network), but some connect to hundreds of targets. For each regulator EDI counts no more than one instance of regulation, whereas GR-EDI considers all. GR-EDI can find patterns that involve only one regulator, such as (0).(hmm.flhD), (1).(hmm.abc_tran) from Table 1. EDI as discussed in [17] would have given a support of one in this case instead of seven for up-regulation (+) and nine for down-regulation (-), and no frequent pattern would have

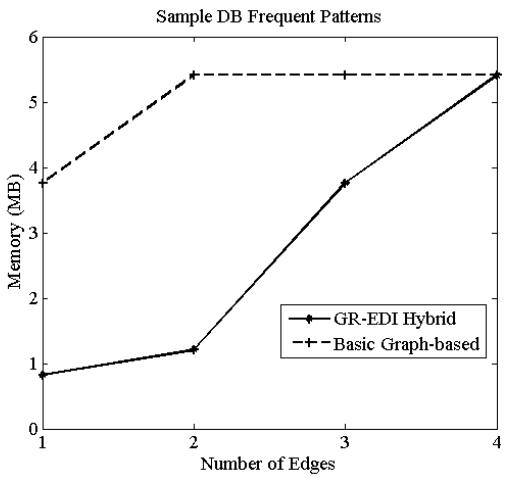

Figure 5: Memory usage for standard Graph-based and GR-EDI hybrid showing the impact of search order on memory.

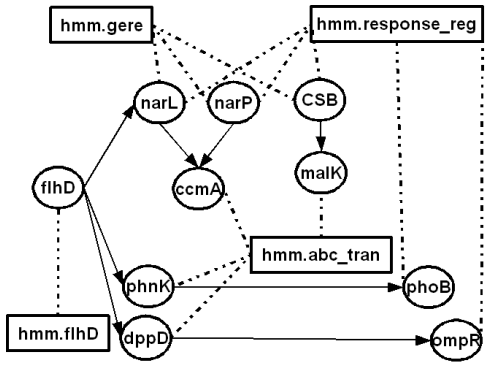

Figure 6: Subgraph from the *E.coli* regulatory network. Patterns related to regulator flhDC and ACB transporter targets were discovered.

Table 1: Example patterns from the *E.coli* network.

Pattern	+ sup., - sup.
(0).(hmm.flhD), (1).(hmm.abc_tran)	7, 9
(0).(hmm.flhD), (1).(hmm.response_reg)	5, -
(0).(hmm.gere), (1).(hmm.abc_tran)	21, 20
(0).(hmm.response_reg), (1).(hmm.abc_tran)	7, -
(0).(hmm.response_reg), (0).(hmm.gere), (1).(hmm.abc_tran)	6, 6

been returned.

These types of graph-relational patterns are of interest to biologists studying global regulators. The pattern shown above suggests that regulators with the flhD domain frequently regulate proteins with the ABC transporter function. Such observations may guide further experiments and may help in interpreting the role of regulators in the context of the functioning of the cell. The biological significance of related patterns is discussed in detail in [2]. Reasons for allowing certain types of overlaps in other biological networks have also been given in the context of network motifs [20, 14, 4].

Figure 6 shows a subgraph of the *E.coli* network and illustrates patterns that our algorithm can discover. Examples of patterns that are detected are listed in Table 1 together with their support in the full network. Note that all of these cases show some level of overlap in the protein properties, but not in the instances of the regulatory connections; therefore the patterns are only found using the GR-EDI mining approach. The patterns demonstrate frequently occurring "modules" or "pathways" that are disjoint in regulatory connectivity. These patterns can be used to relate specific proteins (through their properties) to the global network or to relate global properties to specific locations in the network.

The other major challenge with this type of biological network is the denseness, i.e. presence of hubs that are highly connected. Figure 5 shows the improvement in memory usage for a small sample database based on the *E.coli* set. This random subset was chosen so that both algorithms could complete under reasonable time and memory constraints. For perspective consider that under the full graph the largest 1-unipartite edge pattern under GR-EDI had 10 descriptors requiring the EDI algorithm to wait until 11-edge patterns. The largest unipartite shape discovered using our prototype and memory constraints was size 4. For the sample the largest EDI pattern has 6-edges total and the largest unipartite shape is 4-edges.

It can be seen that although the memory requirements for finding all patterns are the same, GR-EDI returns the results of patterns related to small unipartite shapes with much lower memory requirements than EDI. The order in which GR-EDI visits patterns is such that the inner apriori loop to find results in the same unipartite shape is performed first. EDI, in contrast, visits complex unipartite patterns as it gathers itemset-like patterns for small unipartite shapes. The hybrid search order of GR-EDI has important consequences. Memory usage is minimized for each unipartite shape. That means that once resources are exhausted, results for all smaller shapes are complete. Figure 5 shows that this becomes increasingly important when considering regulatory patterns of more than two proteins.

Having uniform information available at each frequent shape level is very important since patterns that involve a small number of unipartite edges are expected to be much more relevant in biological networks than patterns that extend over many edges. Given the small-world nature of most biological networks, most proteins can be reached by travers-

ing only a few edges. Patterns that are large according to the number of properties, on the other hand, are of great interest since frequent combinations of properties provide additional information for further analysis.

7 Conclusion

In this paper, we present an algorithm for mining frequent patterns in protein networks, in which proteins are associated with multiple properties. The algorithm extends EDI graph mining to multi-labeled settings. We demonstrate that our approach solves several problems associated with the application of existing algorithms to biological networks. The modifications we make with respect to existing approaches are, furthermore, motivated by comparison with traditional single-label approaches. Pattern constraints are introduced to maintain downward closure for support. Our algorithm explores the search space using a hybrid approach that horizontally examines shapes then vertically extends the shapes by adding properties in an apriori fashion. We show that this hybrid approach is particularly suited given the small-world nature of many biological networks. We demonstrate that our algorithm produces meaningful patterns in a real-world example of an *E.coli* transcriptional regulatory network, where conventional EDI produces few or no results.

References

[1] R Agrawal and R Srikant. Fast algorithms for mining association rules in large databases. In *Proc. of the International Conference of Very Large Data Bases*, pages 487–499, Santiago, Chile, Sept 1994.

[2] C Besemann et al. BISON: A Bio- Interface for the Semi-global analysis Of Network patterns. *Source Code Biol Med.*, 1(8), 2006.

[3] Christopher Besemann, Anne Denton, Ajay Yekkirala, Ron Hutchison, and Marc Anderson. Differential association rule mining for the study of protein-protein interaction networks. In *BIOKDD*, pages 72–80, 2004.

[4] J Chen et al. Nemofinder: dissecting genome-wide protein-protein interactions with meso-scale network motifs. In *KDD '06: Proc. of the ACM SIGKDD Int'l Conf. on Knowledge discovery and data mining*, pages 106–115, New York, NY, USA, 2006.

[5] The Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.

[6] L Dehaspe and H Toivonen. Discovery of frequent datalog patterns. *Data Min. Knowl. Discov.*, 3(1):7–36, 1999.

[7] S Ghazizadeh and SS Chawathe. Seus: Structure extraction using summaries. In *DS '02: Proc. of the International Conference on Discovery Science*, pages 71–85, London, UK, 2002.

[8] B Goethals et al. Mining tree queries in a graph. In *KDD '05: Proc. of the ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 61–69, New York, NY, USA, 2005.

[9] Bart Goethals and Jan Van den Bussche. Relational association rules: Getting warmer. In *Pattern Detection and Discovery*, pages 125–139, 2002.

[10] M Halldorsson and J Radhakrishnan. Greed is good: approximating independent sets in sparse and bounded-degree graphs. In *STOC '94: Proc. of the annual ACM symposium on Theory of computing*, pages 439–448, New York, NY, USA, 1994.

[11] H Hu et al. Mining coherent dense subgraphs across massive biological networks for functional discovery. *Bioinformatics*, 21(suppl_1):i213–221, 2005.

[12] A Inokuchi et al. Complete mining of frequent patterns from graphs: Mining graph data. *Mach. Learn.*, 50(3):321–354, 2003.

[13] T Ito, T Chiba, R Ozawa, M Yoshida, M Hattori, and Y Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *PNAS*, 98(8):4569–4574, 2001.

[14] N Kashtan et al. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics*, 20(11):1746–1758, 2004.

[15] M Koyuturk et al. An efficient algorithm for detecting frequent subgraphs in biological networks. *Bioinformatics*, 20(suppl_1):i200–207, 2004.

[16] M Kuramochi and G Karypis. An efficient algorithm for discovering frequent subgraphs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1038–1051, 2004.

[17] M Kuramochi and G Karypis. Finding frequent patterns in a large sparse graph. *Data Min. Knowl. Discov.*, 11:243 – 271, 2005.

[18] Huiying Li, Matteo Pellegrini, and David Eisenberg. Detection of parallel functional modules by comparative analysis of genome sequences. *Nature Biotechnology*, 23:253–260, 2005.

[19] EM Marcotte et al. Detecting protein function and protein-protein interactions from genome sequences. *Science*, 285:751–753, 1999.

[20] R Milo et al. Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

[21] ZN Oltvai and A Barabsi. Life's complexity pyramid. *Science*, 298(5594):763–764, 2002.

[22] T. Oyama, K. Kitano, K. Satou, and T. Ito. Extraction of knowledge on protein-protein interaction by association rule discovery. *Bioinformatics*, 18(8):705–14, 2002.

[23] H Salgado et al. RegulonDB (version 5.0): Escherichia coli K-12 transcriptional regulatory network, operon organization, and growth conditions. *Nucl. Acids Res.*, 34(suppl_1):D394–397, 2006.

[24] P Uetz et al. A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature*, 403:623–627, 2000.

[25] N Vanetik et al. Computing frequent graph patterns from semistructured data. In *ICDM '02: Proc. of the IEEE International Conference on Data Mining*, page 458, Washington, DC, USA, 2002.

[26] N Vanetik et al. Support measures for graph data*. *Data Min. Knowl. Discov.*, 13(2):243–260, 2006.

[27] A Wagner and DA Fell. The small world inside large metabolic networks. *Proceedings of the Royal Society: Bi-*

ological Sciences, 9:1803–10, 2001.

[28] X Yan et al. Mining closed relational graphs with connectivity constraints. In *KDD '05: Proc. of the ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 324–333, New York, NY, USA, 2005.

[29] X Yan and J Han. gspan: Graph-based substructure pattern mining. In *ICDM '02: Proc. of the IEEE Int'l Conf' on Data Mining*, page 721, Washington, DC, USA, 2002.