Generating Abstract Networks Using Multi-Relational Biological Data

Paul Caravelli†, Mitch Beard†, Brian Gopolan†, Lisa Singh†, Zhang-Zhi Hu*
Department of Computer Science†, Department of Oncology *
Georgetown University
Washington, DC 20057
{prc8, msb7, bg227, los4, zh9}@georgetown.edu

Abstract

This paper presents an approach for visual exploration of groups in network data. We let users visually cluster nodes based on common semantic and relational features. We describe the clusters in the context of multi-relational protein data. Finally, we illustrate the clusters as composite nodes using a visual analytic tool and show how to create a meaningful abstracted protein network by connecting these composite nodes based on common membership or common attribute features.

1 Introduction

Modern computing systems are able to generate and store a staggering amount of information. While simple graphs and histograms provide some insight, data with a large number of records or features may still be difficult to analyze and interpret. Visual mining aims to provide coherent, interactive views of large or complex data sets, allowing for user exploration and more rapid understanding of data objects and relationships. Visual mining also allows for a high level of interaction with data and data mining results, enabling users to quickly move between ‘what-if’ scenarios and recursively perform data mining operations on meaningful subsets of the original data.

In this paper, we present an approach for visually exploring groups in multi-relational data. We provide users with different ways to aggregate data based on entity attributes, relationships, and network structural properties. Users can then recursively perform grouping operations and ‘drill down’ across multiple dimensions of the data for exploration and hypothesis generation.

Multi-relational data is inherently difficult to visualize. We choose to model the multi-relational data as a graph or network containing multiple node types and multiple edge types. As users group and cluster data, new abstract networks are created with composite nodes representing different subsets of the original network data and composite edges representing common properties between composite nodes. We choose an abstract graph representation because it is straightforward to interpret and can be used to visually simplify complex data. Our overarching goal is to provide an alternate view of the data that may highlight features that are less visible when viewing the original network, a relational table, or spreadsheet of data.

A number of tools exist for visual exploration of biological networks [4, 5, 11, 6]. Our first contribution is the flexible way we allow users to group nodes. Users are able to highlight any property of the original network, such as protein functions, protein interactions and regulatory pathways in the protein network, and to explore potential hidden relations between parts of the biological system that may provide researchers with suggestions or clues for new biological hypotheses prior to expensive laboratory testing. Our second contribution is an implementation of our grouping approaches in an interactive visual mining tool. Even though the tool is not domain specific, we illustrate an interactive analysis in the context of biological data.

The organization of the paper is as follows. We will begin by describing related literature in section 2. We will then describe the structure of the networks we support in section 3 and discuss our example data set. Our grouping and linking approaches are presented in section 4. Finally, we conclude with future directions.

2 Related Literature

A number of visualization tools exist for exploring biological networks, including Pathway Studio [1] Cytoscape [6], Osprey [4], ProViz [7], PATKA [5], and VizAnt [11]. We refer you to Sudermann and Hallet for a detailed comparison of these tools [9]. Our work is not meant to replace any of these tools. Instead, we are interested in presenting approaches for more sophisticated grouping of nodes and/or edges. While composite nodes exist in some of the mentioned tools, the aggregations are based on commonality of
a single feature. Our aggregations are based on any number of semantic features or network properties and each aggregation can be continuously refined to remove or add detail.

The aggregation of vertices and edges is a well known set of techniques in visual mining, and implementations exist in a number of mining and visualization systems. Wattenberg provides a useful description of existing aggregation techniques, while presenting a new PivotGraph data roll-up approach [10]. PivotGraph rolls up graph data along two axes, and then generates links between elements of those axes, and then generates links between elements of those axes. This provides simple exploration of multivariate clusters. This provides simple exploration of multivariate data. While treemaps could provide a useful description of existing aggregation techniques or edge visual paradigm that users in the biological data. Each node on the outside of the star represents a multi-valued protein attribute or membership. KEGG, PID, and Reactome are three different representations of biological pathways; GOSlim is a cut-down version of the Gene Ontology (GO) in three categories (molecular function, biological process and cellular component). Here, proteins are considered the center of the star because the abstract groupings will focus on grouping proteins based on protein attributes and/or the relationships proteins have to other biological entities, e.g. Reactome and, KEGG pathways, protein functions, etc. We refer to this data model as a multi-relational star. The M*3 network model proposed by Singh et al [8] is a graph-based interpretation of multi-relational data. A multi-relational star is a special case of an M*3 network where one relation is the center or hub of the star (protein) and edges link nodes in the center of the star to nodes that are members of relations on the outside of the star (Reactome and KEGG pathways, protein function, etc.).

More formally, we define our network as \( G = (V_{hub}, V, E) \), where \( V_{hub} \) is the set of nodes representing the center of the multi-relational star, \( V \) is a set of vertex sets representing the perimeter of the star, and \( E \) is a set of edge sets linking nodes in \( V \) to nodes in \( V_{hub} \). Let \( V = \{V_1 \ldots V_n\} \), where \( V_1 \ldots V_n \) are each a set of vertices representing a different entity along the star perimeter. For \( 1 \leq i \leq n \), let \( V_i = \{v_1 \ldots v_m\} \), where \( \{v_1 \ldots v_m\} \in G \). Similarly, let \( E = \{E_1 \ldots E_p\} \), where \( E_1 \ldots E_p \) are each a set of edges. For \( 1 \leq j \leq p \), let \( E_j = \{e_1 \ldots e_q\} \), where \( \{e_1 \ldots e_q\} \in G \). Each edge set relates two vertex sets to one another, such that: \( E_j = \{(v_x, v_y) | v_x \in V_a, v_y \in V_{hub}, V_a \subset V\} \)

Let each vertex set \( V_i \) in \( V \) and each edge set \( E_j \) in \( E \) contain a set of attributes \( \{a_1 \ldots a_k\} \), with every vertex in \( V_i \) or edge in \( E_j \) containing a value for each \( a_k \), for \( 1 \leq k \leq x \). Let any attribute \( a_k \) have possible values \( \{b_1 \ldots b_y\} \). We refer to this set of possible values as the domain of \( a_k \) and denote it as \( \text{Domain}(a_k) = \{b_1 \ldots b_y\} \). Consistent with relational algebra notation, let a particular attribute in set \( V_i \) or \( E_j \) be denoted as \( V_i,k \) or \( E_j,k \), respectively. Let the value of a particular attribute \( a_k \) for a vertex \( v_i \) or edge \( e_j \) be denoted as \( v_{i,k} = b_z \) or \( e_{j,k} = b_z \), respectively, for \( 1 \leq z \leq y \).

3 Background and Motivation

Traditional networks are a succinct and powerful representation for complex biological data. Nodes can be used to represent any entity or concept, e.g. genes, proteins, pathways, cells, disease, and edges can be used to describe relationships between nodes of the same type, e.g. protein-protein interactions, or nodes of different types, e.g. between pathways and proteins, or diseases and genes. This section begins by describing our network model. We then motivate our approach for data aggregation using a known biological network focused on protein related data.

3.1 Network Model Overview

We are interested in multi-relational data that forms a 'star-like' relationship. Figure 1 shows an example for biological data. Each node on the outside of the star represents a multi-valued protein attribute or membership. KEGG, PID, and Reactome are three different representations of biological pathways; GOSlim is a cut-down version of the Gene Ontology (GO) in three categories (molecular function, biological process and cellular component). Here, proteins are considered the center of the star because the abstract groupings will focus on grouping proteins based on protein attributes and/or the relationships proteins have to other biological entities, e.g. Reactome and, KEGG pathways, protein functions, etc. We refer to this data model as a multi-relational star. The M*3 network model proposed by Singh et al [8] is a graph-based interpretation of multi-relational data. A multi-relational star is a special case of an M*3 network where one relation is the center or hub of the star (protein) and edges link nodes in the center of the star to nodes that are members of relations on the outside of the star (Reactome and KEGG pathways, protein function, etc.).
3.2 Data Reduction Using Generalized Blockmodeling

Given that analysis of a biological network may involve hundreds or thousands of vertices connected by even greater numbers of edges, display of all the nodes and edges will be difficult to analyze visually and therefore, have limited utility. Filtering data based on various attributes or network structural constraints can be very useful, e.g. find all proteins annotated with a particular GO term or find all proteins with a minimum number of connections in the network. Sometimes, the task is more exploratory and less targeted. In those cases, it would be useful for the visual analytic tool to run unsupervised segmentation, grouping or clustering algorithms.

Figure 2 is a network with the following types of nodes: proteins (red), GOslim (blue), KEGG pathway (green), PID pathway (yellow), and Reactome (pink) pathways. As in figure 1 proteins are the center of the star. This data set was obtained from the Protein Information Resource (PIR) at pir.georgetown.edu. PIR maintains integrated databases of protein sequences, functional annotations, and family classifications. Although some general information can be gained from the graph as displayed, e.g. a number of proteins show no connections in the network, the visual complexity is far too high for any data analysis to be truly meaningful.

A traditional mathematical aggregate abstraction is the blockmodel or the generalized blockmodel [2]. We define a generalized semantic blockmodel as a generalized blockmodel based on graph semantics instead of graph topology. Instead of using a structural or regular equivalence measure to find clusters, a semantic blockmodel makes use of a measure based on attribute values or relationships within a graph. We believe such an approach could be useful for this protein data set since identifying proteins with common relationships and attributes and attempting to map or infer new proteins to these groups is a central research goal. Unfortunately, one key problem with the use of generalized blockmodels emerges when the user works with graph semantics based on either multiple attributes or multi-valued attributes since creating a block in a blockmodel requires the rows and columns of the model to be partitioned into non-overlapping groups. For a multi-valued attribute, a particular vertex may be a member of multiple groupings, violating the stipulations of the standard blockmodel. Therefore, in the next section we describe an unsupervised grouping approach that is similar to large itemset generation in association rule mining.

4 Semantic Grouping of Multi-relational Star Graphs

Using attribute information associated with the different nodes and edges in the graph, individual vertices can be grouped together if they share the same combination of attribute values. Beginning with the simple case, we can create groups based on a single attribute. Assume each attribute \( a_k \) has domain \( D_k = \{k_1, k_2, \ldots, k_p\} \), where \( p \) is the number of distinct values in \( a_k \). Then each vertex in \( V_{hub} \) is a member of zero or more groups in \( D_k \).

The definition of a semantic group \( g_k \) based on an attribute value \( k_1 \) for a given attribute \( a_k \) is \( g_{k_1} = \{v_i | v_i.k = k_1, v_i \in V_{hub}\} \). Creating all possible semantic groupings for a particular attribute, then, may yield up to \( p \) new semantic groups.

Figure 3(a) shows an example of two sets of overlapping semantic groups, one colored in red and the other colored in blue. The red set of groups is based on vertices in \( V_{hub} \) having specific values of one attribute. In the figure, there are three attribute values for the red attribute - A, B, and C. The nodes in each of those groups represent vertices in \( V_{hub} \) with the attribute value. For example, all the nodes in group A have the attribute value A. The blue groups are based on a second chosen attribute. Similar to the red attribute, the blue attribute has three distinct values - 1, 2, and 3. Vertices from \( V_{hub} \) with those attribute values are shown as nodes in groups 1, 2, and 3.

Visually, each group can be represented as a composite node or a composite edge. For example, we may choose to visually represent each red attribute value as a composite node in an abstracted graph, and each of the blue attribute values as a composite edge that connects composite nodes that have common vertices from \( V_{hub} \) with the same blue attribute value. Figure 3(b) shows an example abstracted...
Figure 3. Example of Semantic Groups

Figure 4. Composite Nodes for Proteins Grouped By KEGG Pathway Linked Using GO Terms to Composite Node Representing Proteins with No KEGG Pathway Information
node in the figure) contain hub nodes with a value of 3 for attribute \( a \) and a value of 2 for attribute \( c \).

For analysis task 4, we are interested in identifying common membership across composite nodes. There are a number of ways to accomplish this visually. We do so, by adding an edge between composite nodes that have the same 'base' members from \( V_{hub} \). We then color the edges based on how 'concentrated' the shared members are to the two composite nodes in question. For example, in figure 6, there are two composite nodes on the left side. One represents nodes having attribute values \( b = 4 \) and \( c = 3 \) (hub nodes \( v_1, v_2 \) in this case), while the other represents nodes having attribute values \( a = 4 \) and \( d = 3 \) (hub nodes \( v_2, v_5, \) and \( v_9 \)).

Because these two composite nodes contain a common member (\( v_2 \)), an edge is placed between them. The color of the edge is based on the strength of the common membership. We calculate the strength based on the number of composite nodes common members appear in. In this case, the edge is a dark red color because \( v_2 \) does not exist in any other composite nodes in the figure, i.e. it has a high shared membership weight. Lighter edges imply that the members of the composite nodes that have shared membership, also have shared membership in other composite nodes, i.e. they have a lower pairwise shared membership weight. Multi-colored edges indicate different weights for different shared members. To avoid edges with a large number of colors, we bin the weights into one of three bins based on the shared membership weights.

Due to space limitations, we do not describe our algorithm. However, because of the sparsity of the data, we use an efficient hashmap structure to generate composite attribute groups given a minimum group membership size and the number of attribute values for each group.

Returning to our biological example, we group proteins based on Gene Ontology term, Reactome and PID pathways. We specify groups of size three and a minimum group membership of eleven proteins. Fifty six composite nodes were generated. We then took the largest composite node created based on three GO terms and linked them to the other generated composite nodes based on shared membership. The results are presented in figure 6. We see that there are only a handful of composite nodes with strong shared membership for all the shared members. Because these are highlighted, researchers can further investigate relationships among these small subsets of proteins.

We pause to mention that all the protein related figures and grouping algorithms described in this paper are implemented in Invenio, a visual mining tool for multi-relational data [8]. Figure 7 shows a screenshot of the tool, highlighting various features. For a more detailed discussion of the

---

**Table 1. Data for Example in Figure 5**

<table>
<thead>
<tr>
<th>( V )</th>
<th>( V.a )</th>
<th>( V.b )</th>
<th>( V.c )</th>
<th>( V.d )</th>
<th>Groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_1 )</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( v_2 )</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>( v_3 )</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>( v_4 )</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>( v_5 )</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>( v_6 )</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>( v_7 )</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>( v_8 )</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>( v_9 )</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>
5 Conclusions

Exploration tasks require a large number of tools. Their strength is evident when pieces of a complex puzzle begin to fit together. This paper proposes an iterative grouping approach for multi-relational data that lets users create groups and break up subgroups for detailed exploration. Groups are created based on topological graph properties and semantic relationships in the data. Our visual simplification of complex data gives users the ability to rapidly identify shared membership across groups and semantic similarity between data objects, thereby facilitating the understanding of clusters that exist in the data.

References


