Visually Comparing a Set of Graphs Dagstuhl Seminar 08191 Working Group Summary

Alejandro Estrella-Balderrama[1], Markus Geyer[2], Carsten Gutwenger[3], Karsten Klein[3], Michael Schulz[4], Mario Albrecht[5], and Oliver Kohlbacher[6]

[1] The University of Arizona, Department of Computer Science
1040 E. Fourth St, Tucson, AZ 85721-0077, USA
aestrell@cs.arizona.edu
[2] Wilhelm-Schickard Institut für Informatik, Arbeitsbereich für Paralleles Rechnen
Sand 14, D-72076 Tübingen, Germany
geyer@informatik.uni-tuebingen.de
[3] Dortmund University of Technology, Department of Computer Science
Otto-Hahn-Str. 14, D-44227 Dortmund, Germany
{carsten.gutwenger,karsten.klein}@cs.tu-dortmund.de
[4] University of Cologne, Department of Computer Science
Pohligstr. 1, D-50969 Köln, Germany
michael.schulz@informatik.uni-koeln.de
[5] Max Planck Institute for Informatics
Department of Computational Biology and Applied Algorithmics,
Campus E1.4, D-66123 Saarbrücken, Germany
mario.albrecht@mpi-inf.mpg.de
[6] Wilhelm-Schickard Institut für Informatik, Div. for Simulation of Biological Systems
Sand 14, D-72076 Tübingen, Germany
oliver.kohlbacher@uni-tuebingen.de

Abstract. We consider methods to visually compare graphs, more to focus on the differences of the graphs than on the similarities. Our two-level approach constructs a meaningful overview of the given graphs combined with a detailed view focusing on a local area of change. The actual layout of these graphs has to be evaluated depending on the specific type of biological network to be visualized in each case. We look into different variants and propose properties to be optimized in our visualizations.

Keywords. graph drawing, visual graph comparison

Dagstuhl Seminar Proceedings 08191
Graph Drawing with Applications to Bioinformatics and Social Sciences
http://drops.dagstuhl.de/opus/volltexte/2008/1553

1 Visualization of Dissimilarities between Multiple Networks

The problem of visualizing differences between families of graphs arises in biology when comparing networks representing different states of an organism (for example, evolution of a disease), time series of a network during a certain process, networks from different organisms, etc. For each of the cases, a set of graphs is given where the difference between two related graphs is small. Graphs from such applications like metabolic networks or protein interaction networks typically consist of several hundreds of nodes, and their structural characteristics (density, degree distribution, etc.) may vary depending on the type of network that they represent. Recent publications show that there is a strong need for better visualizations of network differences, see Figure 1.

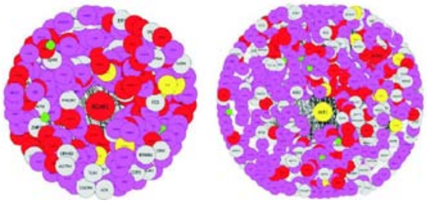

Fig. 1. Example showing two PPI networks for comparison, using a connectivity-based classification of the nodes and a color code corresponding to gene regulation [1].

As a simplification of the general problem, we considered the case where only two graphs G_1, G_2 are given. The changes between the two graphs are assumed to be discrete and relatively small, and the union therefore defines a supergraph $G = G_1 \cup G_2$ which consists mainly of the overlap between G_1, G_2 and a small number of additional nodes and edges that exist in only one of the graphs. Afterwards we tried to divide the necessary work into clearly defined problems that may modularly be combined to an approach to solve the visualization problem. We first developed a general concept describing the user interaction and user interface for a biologically meaningful visualization approach and then derived the problems that need to be solved in order to realize such an interface.

Our approach is based on the observation that the user needs an overview of the whole network structure that allows to recognize both structurally as well as semantically (biologically) relevant features. As the graph is assumed to be to complex to be nicely laid out with regard to the changes between the two graphs, we need some abstraction level that still allows to identify the main features without visualizing the entire graph. Each of the differences should be represented in the overview graph such that position and possible influence may be reasonably estimated by the user. An additional detail view should allow the user to closely investigate areas of change. This can be done by displaying subgraphs of G_1 and G_2 side by side, with a layout that focuses on the differences, see Figure 2.

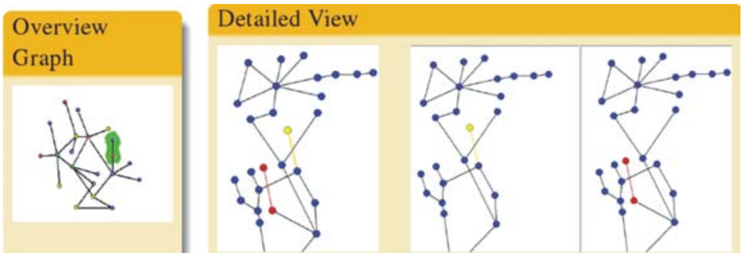

Fig. 2. A schematic representation of the Overview&Detail concept.

The amount of change between these detail graphs is then relatively large in comparison to the differences between G_1 and G_2. We also assume that changes normally have a certain locality, i.e., the affected area can also be drawn in a small portion of the layout area.

We identify the following four main tasks:

1. Backbone construction – construct an overview graph that resembles the characteristics of G.
2. Backbone layout.
3. Detail graph construction – construct a graph that focuses on an area of change.
4. Detail graph layout.

In order to handle the previous tasks, the following problems need to be solved:

- Definition of relevant features that need to be shown in the overview.
- Definition of an area of change - which information is needed in a detail view to allow the user to assess the effect of changes.
- Preservation of the user's mental map for both layouts with regard to well known textbook visualizations.
- Preservation of the user's mental map when changing from overview to detail view.
- Priorities for the two layouts (closely related to mental map issues): Should the local layout be computed first and then the global overview layout be computed to be related to that or should the overview graph be computed first with a relatively independent local layout afterwards? We try to combine the strengths of both approaches by computing local layouts that are optimized to reveal the characteristics of the change together with a neat representation of important substructures, but enriched with context information from the overview graph.

- We also discussed whether a navigation mechanism needs to be incorporated into our approach to be useful for practical applications. We decided that we would like to separate these tasks from one another and did not consider navigational issues any further.

2 Backbone construction

The definition of a helpful backbone is too task-specific to have a single approach that fits all applications. We therefore propose a single instance that we believe might be helpful for at least one practical application (Protein-Protein interaction networks) and design the visualization approach in a modular way such that the other steps are not dependent on specific properties of the backbone computation. Later, the backbone computation can therefore be easily exchanged by another method.

The backbone comprises the following node types:

- **C-nodes** representing areas of change.
- **S-nodes** representing important structures (user- or system-defined input).
- **L-nodes** representing landmarks (e.g. defined by topology).
- **R-nodes** representing remaining parts that may be important even if they do not cover one of the structures above (e.g. large subgraphs).

Areas of change are defined as follows: We call deletion or addition of a single node or edge an *atomic change*. An area of change consists of a single atomic change or an accumulation of adjacent atomic changes, where adjacency means that the graph theoretic distance is smaller than a fixed threshold κ. For our approach, we set $\kappa = 2$.

Edges represent connections between subgraphs corresponding to C, S, L, R nodes and may be weighted. The edge structure should resemble the connectivity patterns in the original graph G, i.e., there exists an edge in B between two nodes if the substructures that they represent are connected in G over a path that does not contain nodes or edges represented by C, S, L nodes in B.

This backbone construction needs to be evaluated using real world data.

3 Backbone layout

Due to lack of time, there was no deep discussion of layout properties or algorithms for the backbone. We propose as a first attempt to use a force-directed approach that may also be guided by application specific information. We leave as future work a less generic approach better suited for specific applications. In general, the main requirements that the backbone layout has to conform are:

- Locality of change – the neighborhood of the C-nodes should not span large parts of the layout area if possible (we assume that this will often be the case with real world data).

- Focus on the S- and C-nodes.
- Incorporate semantical information from the application area.

Further properties may be identified during the evaluation of our backbone construction and simple layout approach using real world data.

4 Detail view

The detail view should focus on highlighting an area of change such that the changes are clearly visible without compromising the overall quality of the layout regarding aesthetical criteria such as edge crossings, etc. In addition, the detail view should allow the user to identify important substructures, either given by topological properties or application-specific semantics. If, e.g., there exist standard drawing conventions for crucial parts of a metabolic pathway (like a vertical downward path ending in a circle), the layout of these parts should resemble the accepted standard. Furthermore, this view should also represent the context, i.e., embed the focused subgraph into a set of landmarks that help the user to keep his orientation in the mental map derived from the overview graph.

4.1 Detail graph construction

A detail graph corresponding to a C-node consists of three main parts:

1. A subgraph of G induced by the expansion of the C-node.
2. The local neighborhood of this subgraph in G.
3. The local neighborhood of the C-node in the overview graph.

There are several parameters that need to be optimized for a specific task or may even be freely adjustable by the user:

- The size of the local neighborhood (e.g. by specifying a graph theoretic distance) in G and in the overview graph.
- The representation of $S-$ and $L-$nodes which may also be expanded or alternatively depicted by a single node possibly with a image label which symbolizes the structure that the node represents.

4.2 Detail graph layout

The detail graph layout should give a clear and concise view of the detail graph (aesthetically pleasing, standard criteria) and at the same time allow to clearly identify the occurring differences. The user should then be able to assess these changes in the larger network context (from a structural as well as from a functional perspective).

We identified the following desirable properties for the visualization of the detail graph (given in descending order of priority):

1. Atomic changes should be drawn nicely (short edges, no crossings, …).
2. The C-node expansion should be close to the center of the drawing if possible.
3. $L-$nodes should be positioned according to the backbone graph layout.
4. Avoid intermixing different subgraphs.

References

1. Anyela Camargo and Francisco Azuaje. Linking gene expression and functional network data in human heart failure. *PLoS ONE*, 2(12):e1347, Dec 2007.