Integrated Visualization of Gene Network and Ontology Applying a Hierarchical Graph Visualization Technique

Rina Nakazawa*, Takayuki Itoh*, Jun Sese+, Aika Terada+
(*)Ochanomizu University, (+)Tokyo Institute of Technology
{leena, itot}@itolab.is.ocha.ac.jp, sesejun@cs.titech.ac.jp

Abstract

A gene network is constructed with genes as nodes, and interactions between genes as edges so as to reveal unknown gene functions and relationship. However, nodes and edges of gene networks are usually very numerous. Because of that, it may be difficult to understand relations between genomic functions and gene-gene interactions, if it is visualized by traditional techniques. This paper presents our technique on visualization of gene networks and gene ontology (GO), which summarizes gene functions and attributes. The technique represents the functions defined by GO as colors of nodes, and bundles edges depending on the gene functions to ease visual complication of the network.

Keywords—**This part is optional.**

1 Introduction

Recent bioinformatics techniques have realized mapping of genetic information; however, still we have many open problems of unexplained genetic functions and relationships. Gene-gene interaction is important information to explain genetic functions and relationships. Gene network consisting of nodes corresponding to genes and edges corresponding gene-gene interactions is a helpful solution to understand them. Network visualization techniques are helpful to understand structures and features of gene networks. However, numbers of nodes and edges in gene networks are often huge, and therefore visualization results with such networks are not often comprehensive.

Gene ontology (GO) is another important information for analysis of genetic functions. GO is an open conceptual system of terms and definitions of genetic information, established to share and utilize knowledge with common vocabulary. Terms and definitions in GO (called "GO term" in this paper) form a DAG (Directed Acyclic Graph). We think that it is fruitful and informative if we visualize the distribution of GO terms on the gene network, because we can discover new relationships between functions defined as GO terms and interactions represented as network. We also think this knowledge may assist prediction of undiscovered genetic functions and relationships, which are important process for independent and customized planning of experiments and analysis of genetic diseases.

This paper presents a technique for integrated visualization of gene ontology and interactions. The technique firstly generates clusters of GO terms according to the DAG structure which arranges the terms, and assigns the term clusters to nodes corresponding to the genes. It then divides the nodes according to the commonality of the term clusters, and divides again according to their connectivity. It calculates the positions of node clusters applying our hierarchical graph visualization technique [9], featuring hybrid force-directed and space-filling approach. The force-directed algorithm attempts to minimize total length and intersections of edges, and the space-filling algorithm completely avoids cluttering of clusters and nodes while it attempts to maximize the space utilization. Finally, it bundles edges connecting pairs of nodes belonging to the common node clusters, and draws the bundles as thicker polygonal lines.

2 Related Work

2.1 Hierarchical Graph Visualization

Hierarchically clustered graphs are an effective data structure for information visualization because they are suitable for overview, zoom, and filtering operations. Several techniques for hierarchical graph visualization apply space-filling techniques such as Treemaps. Zhao et al. [12] presented a technique which interchangeably allows graph diagrams and Treemap to represent parts of a tree. Muelder et al. [10] also presented a Treemap-based technique for hierarchical graph layout.

We presented a visualization technique for hierarchical graphs featuring hybrid force-directed and space-filling methods [9]. The technique assumes that one or more items are assigned to nodes of an input graph. The technique firstly constructs hierarchical clusters of the nodes based on both connectivity and items, and then places the nodes onto a display space. It roughly calculates positions of clusters by a force-directed layout algorithm, so that clusters which are tightly connected or share common items are placed closer on the display space. It then

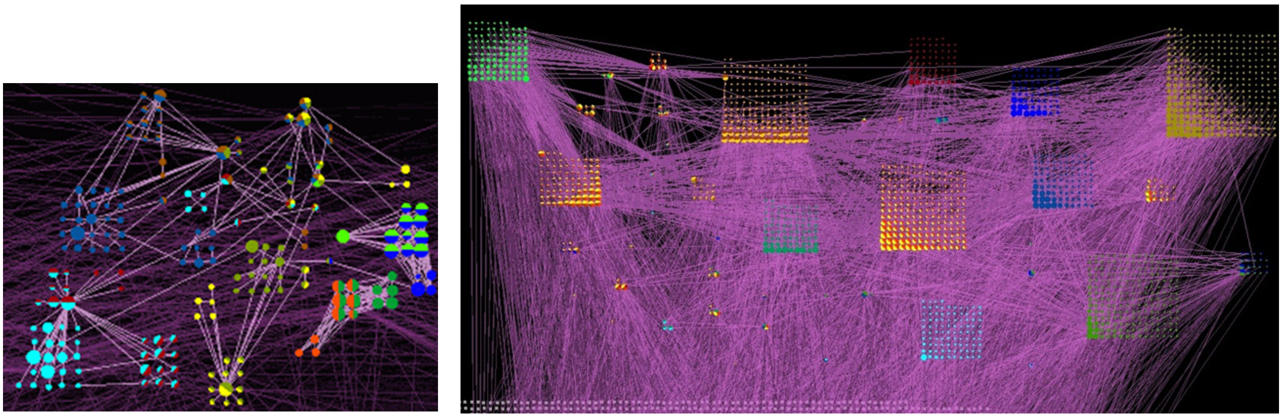

Figure 1: (Left) Example of hierarchical graph visualization shown in [9]. (Right) Problem of gene network visualization observed in the result by our previous implementation of the hierarchical data visualization.

adjusts the positions by a rectangle packing algorithm so that rectangles corresponding to the clusters are tightly packed without overlapping each other. Figure 1(Left) shows an example of network visualization by the technique, where nodes are aligned in rectangle subregions representing clusters, and colored according to their items. The technique can take advantages of both force-directed and space-filling techniques. It attempts to reduce the total length and intersections of edges, as force-directed techniques realize. Also, it completely avoids the cluttering of nodes and clusters while it attempts to maximize the utilization of display spaces, as space-filling techniques realize. In this paper, we adopt this technique to the integrated visualization of gene interactions and ontology information, by mapping the interactions to edges, and the ontology to items.

2.2 Edge Bundling for Graph Visualization

Edge bundling has been a hot topic in information visualization since Holten [7] presented a hierarchical edge bundling technique. The technique hierarchically bundles edges connecting nodes of tree structure level-by-level, and draws them as Spline curves. The curves generally pass through the nodes and clusters, and therefore readability of tree structure may be affected.

To solve this kind of cluttering problem, physically-based approaches that comprehensively draw the bundles while avoiding the nodes are effective. Energy-based [6] [13] or force-directed [8] techniques have been presented, which are especially useful for geographic network datasets because they can attempt to reduce the gap of shapes and places between bundled and original edges. One problem of these approaches is that they often time-consuming for large-scale data. On the other hand, our implementation presented in this paper just bundles edges as polygonal lines while avoiding rectangles corresponding to the clusters of nodes, which realizes smaller computation times.

2.3 Visualization for Gene Network

Several gene network visualization techniques have been presented. Nishiyama et al. [11] presented a technique to visualize network constructed based on gene expression values, which does not represented gene interactions or ontology. Breitkreutz et al. [3] represented gene functionality and network applying variety of standard network visualization techniques, where it seems difficult to comprehensively represent scale-free network datasets.

Our hierarchical graph visualization technique [9] has been applied to visualize relationship and dependency between gene-gene interactions and gene expression values, and actually discovered interesting features including an isolated subgroup of genes which satisfies specific conditions of gene expression values. However, it is not easy to explain the semantics of all features of the visualization results from the knowledge of gene-gene interactions and gene expression values. We think it is often easier to explain by visualizing the functions of genes applying GO terms.

3 Proposed Visualization Technique

3.1 Problem Statement

In this paper we consider input graphs described as $G = \{N, L\}$. $N = \{n_1, ..., n_{n_N}\}$ is a set of nodes corresponding to the genes, and n_N is the total number of nodes. Each of nodes n_i has m-dimensional Boolean values, $n_i = \{b_1, ..., b_m\}$, m is the number of clusters of GO terms, and b_j is *true* if n_i has terms belonging to the j-th cluster. $L = \{l_1, ..., l_{n_L}\}$ is a set of links corresponding to the gene-gene interactions that connect pairs of nodes, and

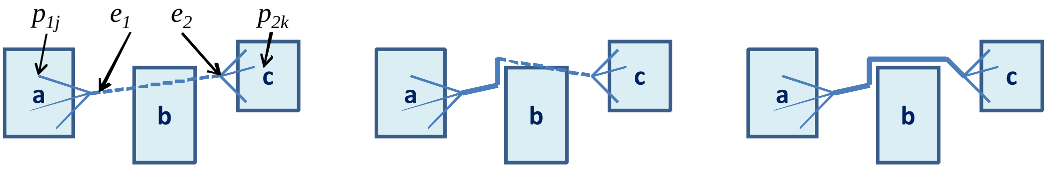

Figure 2: Edge bundling while avoiding rectangles corresponding to node clusters.

n_L is the total number of links. Nodes and links are not weighted.

We prepared a Drosophila gene network dataset provided by iRefIndex [2], containing 8,945 nodes, 32,703 edges, and 259 non-clustered GO terms. We attempted to visualize the dataset by using our hierarchical graph visualization technique [9]; however, we could not generate fruitful visualization results. Figure 1(Right) is an example of the visualization result, which have the following big problems:

- This visualization technique has a serious problem that human vision can distinguish just a limited number of colors. Since it is very difficult for them to distinguish 259 colors, we just assigned arbitrary selected 10 terms as colors of the nodes, and clustered the nodes according to the 10 terms.
- It is difficult to understand relationships among genes because enormous number of edges clutter each other.

Our implementation solves the above problems by the following processes:

1. Divide GO terms into small number of meaningful clusters, and treat the clusters as items of the gene network. Assign distinguishable colors to the term clusters.
2. Divide genes according to items and connectivity.
3. Apply the hierarchical graph visualization technique to calculate the positions of node clusters.
4. Bundle edges connecting two nodes belonging to the same clusters, so that we can easily understand connectivity between node clusters.

The following sections describes the implementation details of the presented technique.

3.2 Gene Ontology Clustering

A set of GO terms are usually used as a DAG (Directed Acyclic Graph). Our implementation marks GO terms used in the set of give genes, and clusters the terms according to the topological distances on the DAG structure. We then extract 10 to 15 term clusters from the clustering results according to specific conditions, such as rougher clustering results, or numbers of genes which terms in a cluster are assigned. We think 10 to 15 is a good number to visually distinguish the terms by the colors of nodes.

3.3 Gene Clustering and Graph Layout

Our implementation applies node clustering as a preprocessing of the graph visualization. It firstly divides nodes to generate groups of nodes which completely the same sets of term clusters are assigned. It then divides the nodes in each cluster according to the density of edges [4].

After the clustering process, the implementation applies two steps of data layout: force-directed and space-filling layout steps. While the space-filling layout step, our current implementation preserves pre-defined distance of blanks between adjacent clusters, so that bundled edges can pass through between them.

3.4 Edge Bundling

Our implementation applies edge bundling after placing all clusters and nodes. It counts the numbers of edges connecting two nodes belonging to arbitrary pairs of clusters It bundles the edges and draws as thicker polygonal lines, if the number exceeds the pre-defined threshold. Otherwise, it does not bundle the edges, and draws them as thinner straight segments. This representation emphasizes major relationships between pairs of genes which common sets of term clusters are assigned.

Our implementation is somewhat similar to Dolulil et al. [5] which avoids passing through rectangular nodes. The implementation firstly extracts a pair of node clusters that edges connecting two nodes belonging to the pair of the clusters. It then generates a segment which connects the centers of the rectangles corresponding to the pair of node clusters, drawn as a thick dotted line in Figure 2(Left). Let the numbers of nodes in each of the clusters which are connected by edges to be bundled as m_1 and m_2, and positions of nodes as $\{\mathbf{p_{11}}, ..., \mathbf{p_{1j}}, ..., \mathbf{p_{1m_1}}\}$ and $\{\mathbf{p_{21}}, ..., \mathbf{p_{2k}}, ..., \mathbf{p_{2m_2}}\}$. Also, let the positions of centers of rectangles corresponding to the clusters as $\mathbf{r_1}$ and $\mathbf{r_2}$. Here, the technique calculates the end points of the segment $\mathbf{e_1}$ and $\mathbf{e_2}$ which connects the rectangles as follows:

$$\mathbf{e_1} = (1-t)\mathbf{r_1} + t\mathbf{r_2}$$
$$\mathbf{e_2} = t\mathbf{r_1} + (1-t)\mathbf{r_2}$$
$$(0 \leq t \leq 0.5)$$

The implementation then detects the collision between the segment e_1e_2 and other rectangles. If the segment intersects with other rectangles, the implementation folds the bundle segment to avoid the intersections, as shown in Figure 2(Center)(Right). Finally, it draws thinner segments $\mathbf{p_{1j}e_1}$ and $\mathbf{p_{2k}e_2}$, and a thicker polygonal line connecting $\mathbf{e_1}$ and $\mathbf{e_2}$.

4 Results

We implemented the presented technique with JDK (Java Development Kit) 1.6, and executed on a personal computer (CPU 2.7GHz Dual Core, RAM 8.0GB) with Windows 7 (64bit). This section shows visualization result with the Drosophila gene network dataset introduced in Section 3.1.

Figure 3(Upper) shows an example of visualization result before applying edge bundling. Interactions between nodes in the upper left cluster indicated as (a) and nodes in the other clusters are well-represented in this example. However, it is difficult to follow gene-gene interactions and major interactions between genetic functions from this example, because of high density of the edges. On the contrary, it is much easier to follow major interactions between commonly featured genes by looking at Figure 3(Lower) which shows a visualization result after applying edge bundling. Here, rectangular regions indicated as (a) to (e) in Figure 3(Lower) are node clusters which nodes are annotated by the following term clusters:

(a): protein-DNA complex (GO:0032993)

(b): intracellular organelle part (GO:0044446),
non_membrane_bounded organelle (GO:0043228),
organelle part (GO:0044422)

(c): non_membrane_bounded organelle (GO:0043228)

(d): cell projection (GO:0042995)

(e): intracellular organelle part (GO:0044446),
organelle part (GO:0044422)

Here, term clusters are described as above by the terms and IDs of the clusters which are placed at relatively higher level on the DAG structures. After applying the edge bundling, we found that there were many edges between pairs of clusters (a) and (b), (a) and (c), (b) and (e), (c) and (e), and (d) and (e). We expect that these major interactions between pairs of genetic functions will be important information of gene network.

Next, we focused on the relationships between two clusters (a) and (b) shown in Figure 4. Here, Figure 4(Left) shows the result before edge bundling, and Figure 4(Right) shows the result after edge bundling. We found that Figure 4(Left) shows two groups of relationships between the two clusters (a) and (b), shown as two arrows. Here, one yellow node at the upper left of the cluster (b) connects with several light green nodes of the cluster (a), whereas it is hard to follow how other nodes of the cluster (b) connect to the nodes of other clusters, while looking at Figure 4(Left). We also observed the same interaction in Figure 4(Right) as well. In contrast to Figure 4(Left), these two relations are brought together, however, we could follow that the cluster (b) had a few-to-many relationship with the cluster (a). According to the one-to-many relationship by the result before edge bundling, we could estimate the relations between other yellow nodes and light green nodes are also one-to-many. Here, the node (b1), at the upper left end of the cluster (b), does not connect with the other clusters, while other nodes (b2), at the lower right side of the cluster (b) connect with. From this result we can assume that the node (b1) has the same interactions as the nodes (b2), has the different GO terms which the nodes (b2) do not have, or other GO terms have an influence on the common relationship.

Our visualization technique can bring this kind of interesting discussions because it effectively integrates the information of genetic functions defined by GO terms and relationships brought from gene-gene interactions.

5 Conclusion and Future Work

This paper presented a new visualization technique for gene network featuring gene interaction and ontology. This technique firstly generates clusters of GO terms according to DAG structure of the terms, and annotate genes the clusters which terms of the genes belong to. The technique then clusters genes according to the GO clusters and connectivity. Finally, it visualizes the clustered gene network applying hybrid force-directed and space-filling node layout and GO-based edge bundling. The paper presented several interesting structural features of the gene network.

Our potential future issues include the following: 1) improvement of edge bundling so that the technique can repulses the bundles completely overlapped on the drawing spaces, 2) evaluation of clustering results, 3) visual analytics with experts in biology, and 4) application of the visualization technique to other fields.

Acknowledgements

We appreciate Prof. Kei Yura of Ochanomizu University for helping us improve this paper. This work has been partially supported by Japan Society of the Promotion of

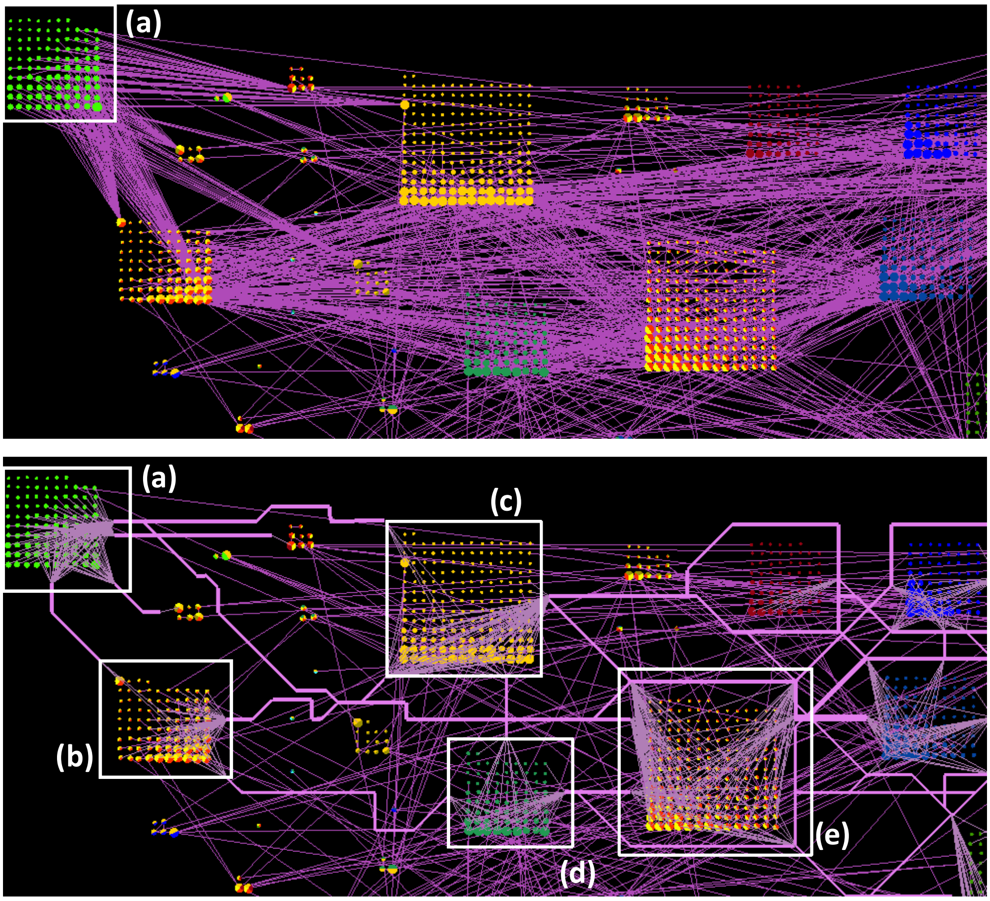

Figure 3: Overview. (Upper) Before edge bundling. (Lower) After edge bundling.

Science under Grant-in-Aid for Scientific Research (C) No. 23500113.

References

[1] The Gene Ontology, http://www.geneontology.org/

[2] iRefIndex, http://irefindex.uio.no/wiki/iRefIndex

[3] B. Breitkreutz, C. Stark, M. Tyers, Osprey: A Network Visualization System, *Genome Biology*, 4:R22, 2003.

[4] A. Clauset, M. E. J. Newman, C. Moore, Finding Community Structure in Very Large Networks, *Physical Review*, E70, 066111, 2004.

[5] M. C. J. Dokulil, J. Katreniakova, Edge Routing and Bundling for Graphs with Fixed Node Positions, *International Conference on Information Visualisation*, 475-481, 2011.

[6] E. R. Gansner, Y. Hu, S. North, C. Scheidegger, Multilevel Agglomerative Edge Bundling for Visualizing Large Graphs, *IEEE Pacific Visualization Symposium*, 187-194, 2011.

[7] D. Holten, Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data, *IEEE Transactions On Visualization And Computer Graphics*, 12(5), 741-748, 2006.

[8] D. Holten, J. van Wijk, Force-Directed Edge Bundling for Graph Visualization, *Computer Graphics Forum*, 28(3), 983-990, 2009.

[9] T. Itoh, C. Muelder, K. Ma, J. Sese, A Hybrid Space-Filling and Force-Directed Layout Method for Visualizing Multiple-Category Graphs, *IEEE Pacific Visualization Symposium*, 121-128, 2009.

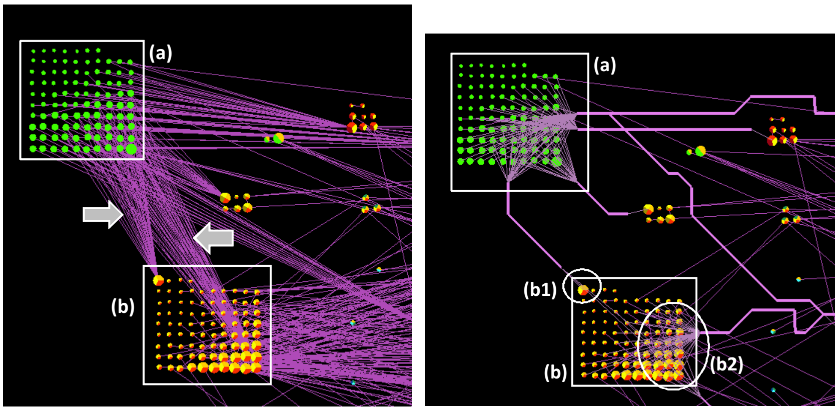

Figure 4: Close up view. (Left) Before edge bundling. (Right) After edge bundling.

[10] C. Muelder, K.-L. Ma, A Treemap Based Method for Rapid Layout of Large Graphs, *IEEE Pacific Visualization Symposium*, 231-238, 2008.

[11] K. Nishiyama, T. Itoh, Visualization of Hierarchical Gene Network using HeiankyoView, *The Journal of Society for Art and Science*, 6(3), 106-116, 2007. (in Japanese)

[12] S. Zhao, M. J. McGuffin, M. H. Chignell, Elastic Hierarchies: Combining Treemaps and Node-Link Diagrams, *IEEE Information Visualization*, 57-64, 2005.

[13] H. Zhou, X. Yuan, W. Cui, H. Qu, B. Chen. Energy-Based Hierarchical Edge Clustering of Graphs, *IEEE Pacific Visualization Symposium*, 55-61, 2008.