GraphLinker: A Visual Comparative Environment of Genomic and Metabolic Networks

Niels Hanson*
University of British Columbia

ABSTRACT

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

1 INTRODUCTION

Microbacterial communities make up a wide and diverse environment which accounts for some of the most unexplored biomes in the world. Metagenomics, the study of the genomic information contained within a community, represents a window with which to view and explore the diversity and dynamics of these naturally occurring communities. One of the new emerging concepts in this field is the distributed nature of metabolism between many different microorganisms within an environment. Metabolism, the life-sustaining chemical reactions that enable an organism to survive, in higher-order organisms are often viewed in isolation of other species, however, it is becoming increasing apparent that these metabolic networks in microbacterial communities are distributed and shared between a number of different organisms in the environment. Organisms rely upon, and optimize themselves with respect to others making up a macroscopic biological system. By acknowledging this macroscopic view and taking a Systems Biology approach, we can investigate how microorganisms are sharing their metabolism within the community and how that community changes over time.

To add some formalism to the problem, we can treat microorganisms (taxa) and metabolic pathway steps as nodes in two separate, though intrinsically related, networks or graphs. The presence of a specific taxa can be estimated in a sample from the environment by sequencing genomic material, specifically small ribosomal subunit RNA (16S RNA), a section of the bacterial genome that is well suited for identification. A collection of sequences are clustered by multiple sequence alignment into operational taxonomic units (OTUs) which correspond to one or more closely related microorganisms in a sample. Metabolic reaction steps exist in a pathway of known overall metabolism with a complex yet defined order of substrates and products representing the inputs and outputs of each reaction. The presence or absence of these enzymatic steps can be inferred via functional genes found in the sequenced environmental genomes. The PathoLogic algorithm developed by the International Stanford Research Institute (SRI) can infer the presence or absence of specific pathway steps, effectively supplying nodes, their connecting edges, and their weights in the metabolic graph [14]. OTUs can be clustered in a hierarchy using correlation algorithms, which essentially describe edges and weights within a corresponding taxa graph [15]. The problem is of course to effectively infer the influence of the taxa graph on the metabolic one, which essentially boils down to a version of the Graph Isomorphism problem, a classic NP problem in computer graphing.

Here we present GraphLinker, an exploratory environment for the visual comparison of two graphs via their weighted associations. Gathering inspiration from other graph visualizaton studies, we use a force-directed layout to take advantage of the positional visual channel, and a specialized edge encoding to highlight significant meta-edges, edges that connect nodes from one graph to the other. We validate the method for use with both a random graph, as well as a processed metagenomic sample, highlighting taxa present, taken from the waters of the pacific west coast of British Columbia.

2 RELATED WORK

Problems involving graphs and their visual interpretation have in some ways existed since the time of Euler. However, the formalism of graph visualization community really grew around yearly Symposia on Graph Drawing which started in 1992 in Rome. Since then, research has identified three general overlapping problems: node and edge occlusion by density, readable edge layouts, and computational complexity in graph drawing [10]. The intuitive impact of graphs and their cognitive interpretation has unfortunately been less formally studied when compared to other aspects of visualization like colour. This makes the area more holistic and reliant on individual usability studies for validation [10].

Many real world graphical datasets, including most biological ones, have the global property of a large number of highly distributed clusters with a small average path-length. This is known as the small-world or power law property in the literature and has been be shown to occur many different datasets across multiple knowledge domains, including many biological ones. Recently, there have been a number of attempts to optimize the visualization to graphs having these properties though the specialized use of force-directed layouts [6]. These forced-based approaches, though initially too computationally complex and non-deterministic, have had a number of algorithmic optimizations that make the problem tractable [10]. More recent developments have been to apply force-directed approximations to online (live) graphs to facilitate user interaction [7]. Ham and Wijk have used both semantical and geometrical distortions to create scalable, interactive visualizations of small-world graphs [17].

Another aspect of graph layout has been the visualization of hierarchical clustering. A number of classic layouts are highlighted by Herman in his review, however, the work of Archambault et. al. has investigated into the visualization of small-world graphs from a number of additional angles. In TopoLayout he proposed a multilevel algorithm that draws undirected graphs based on the topological features and improves upon a number of different layout algorithms [5]. Grouse, another visualization environment, tackles the hierarchical normally shown in a tree layout by collapsing nodes into concentric meta-nodes with meta-edges and raised the visualization of the graph up the structural hierarchy [4]. Furthermore, Archambault has recently touched upon graph comparison through the overlapping of two graphs in difference maps when nodes of the two graphs being compared exist in the same set of objects [2].

These are all useful and relevant advances, however, to our knowledge our proposal represents the first attempt at the visual

*e-mail: nielsh@interchange.ubc.ca

mapping of two related graphs from different domains. As far as mapping edges in general, Graph Isomorphism is one of the twelve classic computational complexity problems proposed by Gary and Johnson back in their 1979 seminal textbook of the subject [8]. However, it is peculiar in that it remains one of the last to still have its computational complexity unsolved. It is still unclear if the problem exists in the set of polynomial or NP-complete problems [12]. The best algorithm to date runs in $2^{O(nlog(n))}$ where n is the number of vertices in an undirected graph [11]. However, despite not having the complexity of fan exact solution solved, several practical algorithms have been proposed which in most cases run in polynomial time despite being still exponential in the worst case [16].

Bringing the problem back into biological context, previous attempts to solve a similar mapping of gene presence to pathway targets have been attempted, however with mixed success, classifying less than 50% of enzymatic genes to a pathway [9]. This could be due to a number of factors, including deficiencies in the KEGG pathway database. However, these previous attempts considered bacterial taxa in isolation as opposed to a Biological System which would accommodate the distributed metabolism amongst many taxa. It is becoming increasingly apparent that this latter approach is may be the more appropriate perspective to take.

3 Proposed Solution

The two graphs being compared are drawn in the same view, separated spatially at opposite ends, with the separate nodes encoded by two distinct colours. Nodes in each graph were separated by applying a force-directed layout prior to the inclusion of edges between the graphs. Meta-edges, weighted edges that connect nodes between the two graphs, were redundantly encoded by three visual channels of colour, transparency, and edge width (Figure 1). Edges within graphs were kept at a standard width and color as the goal of this visualization is to highlight connections between graphs through the weighted meta-edges. In addition, as any node in one graph could potentially connect with any other node in the other, this would inevitably lead to a large number of edge crossings, so a prominent visual encoding was required to make sure these edges were not lost in the masses of others. Colours were chosen to be suitable for a continuous gradient and to be compatible for individuals with colour blindness [3].

4 Implementation

This implementation of the solution used Cytoscape (Version 2.8.x), a graph viewing visualization environment popular with the biological community, and was developed as plugin. This was preferable as numerous other biological plugins have been developed, in addition to having an environment that is optimized to handle very large graphs and multiple layouts implemented [1]. Our plugin was implemented in java using the Cytoscape plugin API[13]. It consisted of two major methods, one that draws graphs from three input files specifying nodes of both graphs and their connections, and another that specifies the visual encoding specific to the node and edge attributes.

The force directed layout algorithm implemented internally into Cytoscape was used as is, however, it is applied first to the two graphs to be compared without connecting meta-nodes to separate the graphs. We tried a number of other ways to fool the force-directed algorithm into separating the two graphs, including creating many low-weight invisible edges the graphs. This worked well for a small number of nodes, however, since we need an invisible edge from every node in one to every node in the other, the additional edges taxed the force-directed algorithm to the point where it became intractable. The first approach of applying the force-directed algorithm to the two graphs prior to adding meta-edges is imprecise, but it adequately separates the two graphs enough that it did not hinder finding meta-edges and doing comparisons.

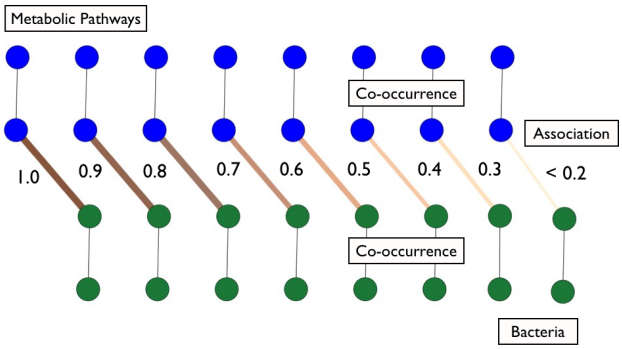

Figure 1: An outline of the visual encoding in GraphLinker. Microorganisms and their co-occurrences are represented by the green nodes and standard thin edges. Metabolic pathway co-occurrences are represented by blue nodes and standard edges. Meta-nodes, weighted un-directional edges that connect the two graphs, are redundantly encoded by three separate visual channels: colour gradient, transparency, and edge width. Meta-edges are the wides and most salient when the underlying association is strong, near 1.0, and subsequently decreases when the association gets to weak, the current visible cutoff being near 0.2

More detail on the specific usage of the plugin can be found in the *README* file of the *GraphLinker.zip* package.

5 Results

GraphLinker was validated on three use cases. The first is a trivial diagnostic case that is used to ensure that the software was properly loaded into Cytoscape to showcase the visual encoding, and to test for force directed separation of highly disconnected graphs (Figure 2). The second is a 100 x 100 node random graph with 200 random edges to test the separation of the two visual encoding for occlusion and the ability to discern significant connecting edges between the two graphs (Figure 3). The third, is a realistic metagenomic sample taken from the Line-P ocean time-series containing 622 nodes and 832 edges (Figure 4). It also follows the small-world property to some degree. In the worst use case our metagenomic samples would be approximately 1000-2000 taxa against hundreds of potential pathways, which makes this test case a little on the small side.

Notice in the random graph use case that despite having many overlapping edges connecting the two graphs we can still discern the most significant edges due to the combination of visual encodings use to encode them (Figure 3). This is an encouraging result suggesting that the combination of a linear colour gradient, transparency and edge width is an effective visual encoding and ensures that significant edges can be seen above the crowd.

The third Line-P example represents our realistic test case. Notice that we can still locate significant connections between the taxa graph (green) and the metabolic pathway graph (blue) by their strong visual encoding. (Figure 4) Looking up these significant edges we notice that these are indeed clustering with anaerobic bacteria and sulphate reduction, two key indicators that this sample came from a low oxygen environment. This is an encouraging result as we can potentially start to follow the patterns of connections between the two networks; the goal this tool aimed to accomplish.

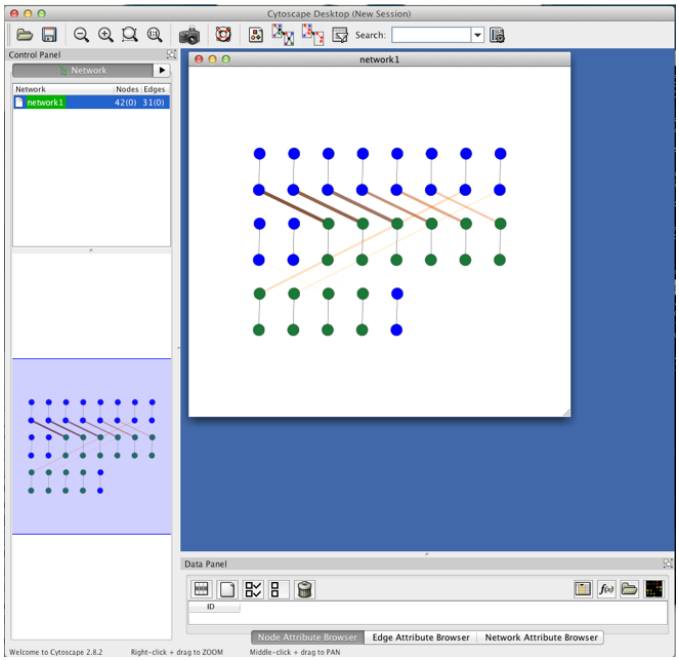

Figure 2: A simple diagnostic plot showcasing the visual encoding, the force-directed layout, and ensuring that the Cytoscape plugin was loaded correctly. One can notice the full spectrum of the meta-edge encoding through 0.1 intervals from 1.0 to 0.0.

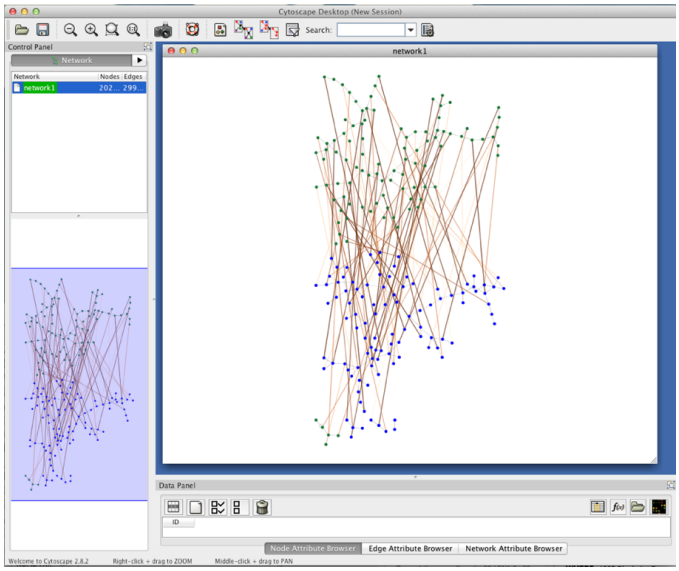

Figure 3: This 'random' data set of a 100 x 100 graph with 200 random edges is used to test the resilience of the visual encoding when in areas of very dense edges. We can see that strong connecting meta-edges can still be seen relatively easily despite the edge crowding.

6 Discussion and Future Work

This initial implementation is encouraging in its initial results that the visual encoding works to some degree, however, it does not meet all the goals initially set out in the project proposal. There are four major areas where things could be improved to highlight comparisons between graphs: edge-management, graph separation

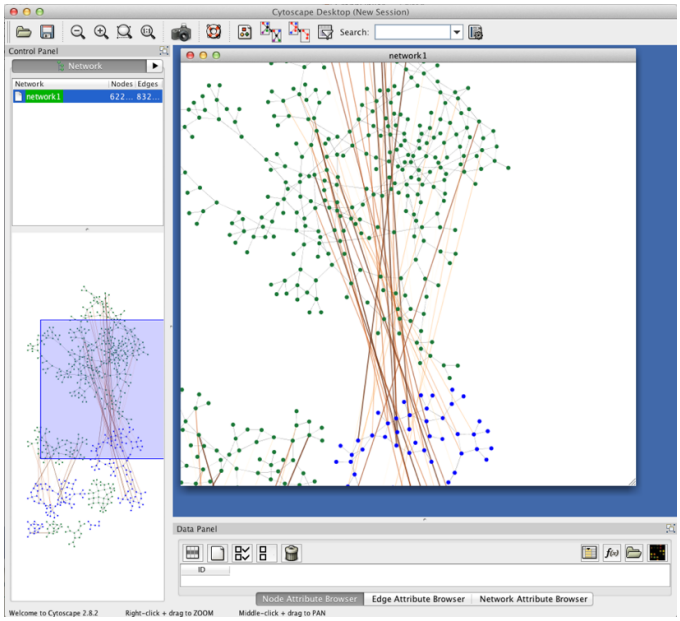

Figure 4: An example of a realistic case from a metagenomic data set. This dataset consists of approximately 800 nodes with 100 connecting edges between the two graphs. We can identify strong connecting edges between the taxa graph (green) and the metabolic pathway graph (blue) by their strong visual encoding. Upon looking up these strong edges we can find clusters of methanogenic and sulphur-reducing bacteria, something that we expect to find in low-oxygen environments.

and layout, hierarchical clustering, and general usability/Cytoscape integration. It is best to take this project as a proof of concept rather than a fully-fledged visualization environment. Many of these improvements are needed to make this jump but are not essential to making a novel contribution to showing relations between graphs visually, which is what was focused on.

Like just about all graph visualization projects, one must confront the vicious problem of edge crossings, as it is the number one factor that makes them difficult to read [**?**]. In GraphLinker, important connecting edges were highlighted by a combination of three visual encoding channels, however, this is a more practical than elegant solution to the problem. There are a number of different approaches to solve this problem. Finding a layout that minimizes edge crossings is known to be NP-complete, though approximate solutions though rotational and tension based models are known to be tractable [**?**]. Edge bundling is another alternative been shown to be effective. Additionally, edges would be reduced by collapsing nodes into a hierarchical structure as seen in the Grouse visualization environment by Archambault *et. al.*. A combination of all of these would likely be ideal, the use could click to expand and contract the meta-node hierarchy and at the same time reduce the number of edges incident by bundling.

In GraphLinkers implementation, we staggered the application of the force-directed layout to get the partial separation of the two graphs, as an absolute separation between the graphs using hidden edges proved intractable. An alternative solution is to apply the algorithm to each graph individually and then separate the graphs to opposite sites of the view. However, this layout doesnt take into account the connecting nodes, it would be good to find a layout that highlights them in some ordered manner. A layout that reorganizes the graphs this way would also minimize edge crossings by arranging nodes such that meta-edge distance is minimized. Such an algorithm may be implementable in Cytoscape though given the

limited existing documentation, its depreciating API, and the impending overhaul of the structure with the coming update to 3.0, it would be best for the Cytoscape developers to make their changes before any additional development.

The collapsing of nodes into hierarchical clusterings would definitely be a good development in reducing the number of onscreen nodes and the number of edge crossings. A sketch of this was shown in the project proposal, taking inspiration from the work of Archambault et. al. in the Grouse visualization environment (Figure 5). This was not approached in this iteration because it is unclear if it could be implemented in Cytoscapes current structure. It is unclear if containment areas could be fashioned to contain nodes, as well as be meta nodes themselves. There is also limited ability to hide and show existing nodes in a graph, which would be paramount in achieving this functionality.

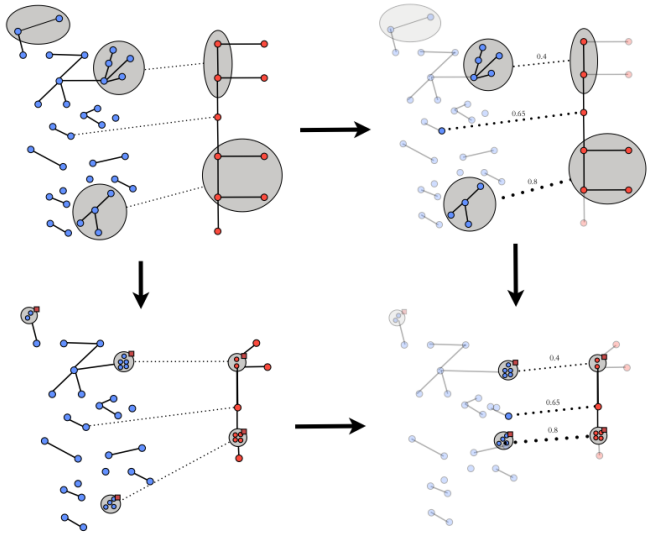

Figure 5: Original Ideas from the Project Proposal. In the project proposal an idea of encoding hierarchical clustering as proposed to reduce clutter and increase salience of connecting meta-edges, taking strong influence from the work of Daniel Archambault *et.al.*. It is unclear if this can be implemented in the current Cytoscape (v2.8.x) API and schema.

Finally, the current usability of GraphLinker is complied from the almost non-existant by todays standards. It is compiled from the command-line, with input files in specific positions with specific names (see *README* file in *GraphLinker.zip* package). In order to turn this into a fully integrated plugin to Cytoscape, it would have to be fully integrated with the file loader and the local node and edge attributes displayed. I decided against this again, because of Cytoscapes impending upgrade. Overdeveloping at this stage would be wasteful if the entire structure of the program is going to change.

To conclude, GraphLinker represents a first step to a comparative graph visualization tool, and can be taken as a proof-of-concept of the visual encoding and applicability of the problem. Future developments may or may not continue in Cytoscape depending the capabilities of the 3.0 update, and so may to a different visual environment for development.

References

[1] R. E. W. H. Aaron Barsky, Jennifer L. Gardy and T. Munzner. Cerebral: a cytoscape plugin for layout of and interaction with biological networks using subcellular localization annotation. *Bioinformatics*, 2007.

[2] D. Archambault. Structural differences between two graphs through hierarchies. *Proceedings of Graphics Interface*, 2009.

[3] G. W. H. Cynthia A. Brewer and M. A. Harrower. Colorbrewer in print: A catalog of color schemes for maps. *Cartography and Geographic Information Science*, 2003.

[4] T. M. Daniel Archambault and D. Auber. Grouse: Feature-based, steerable graph hierarchy exploration. *IEEE-VGTC Symposium on Visualization*, 2007.

[5] T. M. Daniel Archambault and D. Auber. Topolayout: Multi-level graph layout by topological features. *IEEE Transactions on Visualization and Computer Graphics*, 2007.

[6] F. J. David Auber, Yves Chiricota and G. Melancon. Multiscale visualization of small world networks. *IEEE Transactions on Visualization and Computer Graphics*, 2003.

[7] Y. Frishman and A. Tal. Online dynamic graph drawing. *IEEE Transactions on Visualization and Computer Graphics*, 2008.

[8] M. R. Garey and D. S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. W.H. Freeman and Company, first edition, 1979.

[9] W. F. H Ogata and S. Goto. A heuristic graph comparison algorithm and its application to detect functionally related enzyme clusters. *Nucleic Acids Research*, 2000.

[10] G. M. Ivan Herman and M. S. Marshall. Graph visualization and navigation in information visualization: a survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(X):XX–XX, June 2000.

[11] D. S. Johnson. Computers and intractability. *ACM Transactions on Algorithms*, 2005.

[12] W. Mulzer and G. Rote. Minimum-weight triangulation is np-hard. *Journal of the ACM*, 2008.

[13] O. O. N. S. B. J. T. W. D. R. N. A. B. S. Paul Shannon, Andrew Markiel and T. Ideker. Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome research*, 2003.

[14] M. K. M. L. J. M. D. T. J. L. P. K. F. G. A. S. L. P. T. A. I. P. I. M. K. Peter D. Karp, Suzanne M. Paley and R. Caspi. Pathway tools version 13.0: integrated software for pathway/genome informatics and systems biology. *Briefings in Bioinformatics*, 2(1):40–79, Aug. 2009.

[15] J. P. Samuel Chaffron, Hubert Rehrauer and C. von Mering. A global network of coexisting microbes from environmental and whole-genome sequence data. *Cold Spring Harbor Laboratory Press*, 20(1):947–959, Aug. 2010.

[16] D. C. Schmidt and L. E. Druffel. A fast backtracking algorithm to test directed graphs for isomorphism using distance matrices. *Journal of the ACM*, 1976.

[17] F. van Ham and J. J. van Wijk. Interactive visualization of small world graphs. *IEEE Transactions on Visualization and Computer Graphics*, 2004.