DataViz: Visualizing Metabolic Pathways

Nereida Aguilar, *Dominican University*
Ike Anyanetu, *University of Michigan*
Megan Wilson, *Clemson University*

Abstract

Tools for visualization of biological data focus more on the algorithms behind data processing than on the needs of their biologist users. Scientists in the field of biology and bioinformatics require software that is more versatile, yet simple to use. This has become a necessity as the depth of information scientists have to organize becomes more and more complex. Disparities between the programs capabilities and their users' needs lead to problems including difficulty in importing data, exporting data, and data manipulation. These problems are demonstrated by the inability for many bioinformatics tools to adapt to quickly changing user needsthis ranges from a wider aggregate of biological databases to more intuitive representations of biological data. We seek to focus on the latter, evaluating limitations in visualization software to better fit user needs. One such problem is the way metabolic flux is visualized in the widely used graph visualization program Cytoscape. We investigate different node-edge representations using prototypes of varying edge shapes and colors to denote flux magnitude and direction. Our aim is to explore different methods of visualizing graph edges, providing an effective analysis for portraying graph data in a clear and concise manner.

I. Introduction

Technology has led to new laboratory techniques, leading to the generation of large amounts of data in the biological sciences. As the amount of information available to scientists increases, the abilities of current technology needed to accommodate these changes must improve accordingly. Existing technologies are largely focused on the processing algorithms behind the data than the needs of their most common users. These users, primarily scientists focused in the biological sciences, require software that is more versatile, yet simpler to use. Usability tests have shown that while biological analysis through the use of software tools is becoming increasingly common, only general exploratory analysis is truly supported [1]. Mirel suggests that there needs to be

more functionality in the programs such as querying and visual modeling.

One of the largest issues in the field of data visualization is the need to efficiently bring a variety of biological networks into formats that can be clearly interpreted and easily understood by users. Previous investigations by biological researchers concluded that there are components of pathway system representations that are key to effective data visualization [2]. Included in this list is the ability to overlay information on the pathways in a visually meaningful manner and enabling users to see multiple interconnections simultaneously [3].

One method in which current software can be improved on is the way data is displayed on an edge in a standard node-edge network. In the field of biology and bioinformatics, reactions between molecules and enzymes are represented as edges between nodes in a graph [4]. Attributes such as reaction rates, stoichiometry, and flux data are examples of data that are viewed on a graphs edges, allowing the user to easily determine the magnitude and/or importance of a specific reaction in a pathway. The current method of edge visualization is to use a node-link representation using a standard arrow pointing to the target node to show directionality. Attributes of the interactions of nodes are typically shown as a number labeled on the edge or as a node in the middle of the reactionthis is typically the magnitude, or empirical value of the reaction (a standard representation of a network is shown in Figure 1).

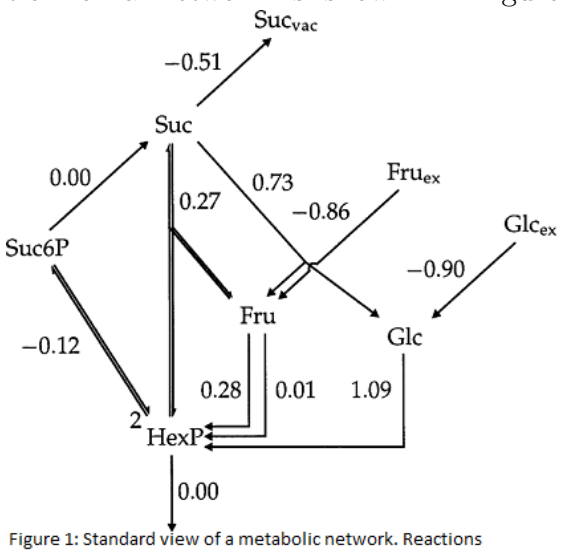

Figure 1: Standard view of a metabolic network. Reactions between molecules are shown with arrows and numerical values depict flux magnitude (negative values indicate flux values occurring in opposite direction).

II. Related Work

1 Background Information

In order to understand the abilities and limitations of the current software, we conducted a few interviews with researchers in the field of biology and bioinformatics. Our selection of interview participants included faculty and graduate students at Iowa State University including: Dr. Julie Dickerson, Dr. Eve Wurtele, Dr. Jacqueline Shanks, Jong Moon Yoon, and Ting Wei Tee.

Dr. Julie Dickerson, associate professor in Electrical and Computer Engineering

at Iowa State University, whose primary research is in the area of bioinformatics, claims that software developers need not do much to improve current visualization tools. The ability to handle larger graphs, but with more functionality is her biggest concern for visualization programs like Cytoscape. Dr. Jacqueline Shanks, Manley Hoppe Professor in chemical engineering and plant systems engineering, is focused on the reliability of current programs. Software that is hard to use and provides a representation of data that isnt meaningful is her biggest concern. She envisions that programs should treat biological pathways like cars in traffic the location of each molecule and enzyme in a pathway should grant the user information about the system. Dr. Shanks also states that different levels of abstraction for viewing these interactions is important, as well as gaining as much or as little detail as necessary for a researcher to complete a task. This concept of program flexibility is also supported by Dr. Eve Wurtele, a professor in genetics and cell biology. Dr. Wurtele emphasizes the importance of program versatility, claiming that research on data visualization in plant molecules needs tools that can collapse large networks to a simpler viewa problem only recently acknowledged by these programs [3]. The necessity for increasing program capability without increasing complexity is also noted by Yoon and Tee. They mention that expanding user control as much as possible is ideal, but not if the implications involve large, bulky programs that are hard to navigate.

To establish a starting point for our software analysis, we investigated some of the most prevalent tools for biological graph visualization. Included in our search are the programs Cytoscape, VisANT, NetworkX, and PathCase. Cytoscape, originally created for biological research, is now a common platform for network analysis and visualization [5]. It allows users to create plug-ins for any large modification they want to see within the program. VisANT is an visualization application that provides many different interfacesa completely online, but limited, web application, a downloadable version that allows for customization, and a large version or very large networks [6]. NetworkX, a software package for Python, provides much needed tools for analysis of networks and visualizations of their physical layout [7]. Despite its completeness and functionality in being able to modify the way networks are laid out, NetworkX provides only a mechanism for visualization. The materials necessary to produce graphs of any significance must be provided by the user. The PathCase software is more biologically based than the previous few. It features a web-based system, an interactive metabolic pathway tool for clients, and the ability to query in many different ways [8].

2 Software Analysis

In order to evaluate the different pieces of software we created the following set of criterion based on biologists feedback and Saraiyas five important features of software

for exploratory pathway analysis [3]:

Web-based

The ability for each program to be web-based is a growing concern as biological databases are mostly online. Requiring the user to download and install an interface is an obstacle that would be ideal to circumvent.

Import/Export Files

As common power tools need drill bits and other pieces to implement their jobs, many biological visualization tools require pieces to function fully, such as the uploading of a pathway spreadsheet, or converting a pathway graph to an image file. This includes data input/output of the users choice.

Layout Modification

Arguably the most important feature of a widely-used piece of graph display software, the control a user has over his/her environment is crucial. There are countless ways in which a user may want to manipulate a pathway, including modifying node/edge size, color, and shape. There may also be a desire for the user to create original modifications to be implemented.

Updated Information

Although this feature usually comes with the Web-based feature, some offline programs do not have an up-to-date database of biological pathway information. Some programs use direct links to multi-organism databases such as KEGG, MetaCyc and BioCyc to keep their records relevant, but that is not the case for every program [9].

Command Line Interface

The ability for each program to have a command line interface is not as important as the previous four criteria, but its still useful for both batch processing of data, and simple interfacing from outside of the programs interface. Table 1 describes how well each of the programs fit the above criteria, leading us to land on Cytoscape as a starting point for feature modification.

Program	Web-based	Import/Export Files	Modify Layout	Up To Date (Biological Databases)	Command Line
Cytoscape	✓	✓	✓	✓	✓
VisANT	✓	✓	✓	✓	✗
NetworkX	✓	✓	✓	✗	✓
PathCase	✓	✗	✓	✗	✗

Table 1: Table of software criteria

Cytoscape, originally released in 2003 as an open source platform, is the most widely-used tool for biological pathway visualization [5]. It has gone through numerous revisions since then and has a simplified plug-in structure to allow for user modifications. Its most prominent feature is the large developer base that has enabled a variety of plug-ins to be developed for varying purposes [4]. The basis for our user study focuses on how metabolic flux is visualized in this software.

3 Creating a User Study

Metabolic flux describes the rate of enzymatic conversions between molecules in metabolic pathways. There are currently very few existing techniques to visualize flux values, despite the growing need for flux data to be represented graphically [4]. The most prominent Cytoscape plug-in for

visualizing flux, by Konig and Holzhutter, is called FluzViz [10]. It can import networks of specifically designed to represent flux values, as well as export flux visualizations to common output formats, including PNG and PDF formats. It is limited, however, by the inability for the user to quickly and clearly modify the layout of the software. Upon loading a network, the user is presented with a black-and-grey view of the network, with one of the few new visualization options being black lines that use thickness for magnitude, and an arrow for direction (Figure 2).

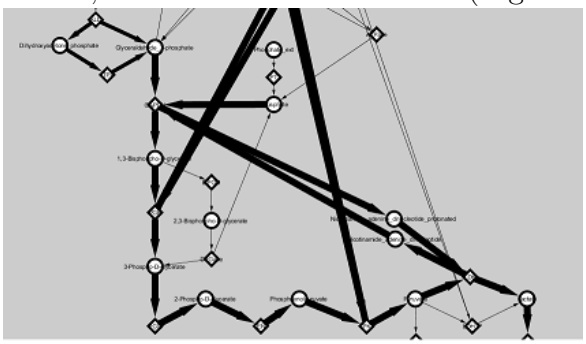

Figure 2: FluxViz uses arrows to denote flux directionality, and thickness to denote flux magnitude.

We would ideally like the user to be able to customize the visualization of this information, and a number of solutions have been proposed. In their study on how people perceive graphs, Holten and van Wijk found that when it comes to nodes and edges, arrows are not best graphical representations of direction. They showed that tapered edges, fading from light to dark/dark to light, and color fading were all more effective than arrows when it came to correctly determining edge direction [11]. We seek to expand on their findings by studying how well users can determine magnitude of an edge, as well as direction. This additional feature is necessary in the field of bioinformatics, for biologists are interested in not just if two nodes interact, but how they interact. The ability to quantify these interactions is also growing concern in the field [12]. To examine how users understand the representation of both edge direction and magnitude, we conducted a survey they separates several different ways that graphs can be represented. In his study on visualizing graphs with directed edges, Holten showed that arrows do not necessarily give the clearest depiction of edge direction and magnitude [11]. We based our prototypes on this premise, as the way a user interprets optimal visual representation is, for the most part, subjective. Some of the prototypes Holten used in his survey laid the foundation for ours. We also took Beckers analysis of graph alternatives into account. He found that in large networks, the common line-arrow representation with a numerical value to denote magnitude is not always the most clear to a user [13]. Taking this into account, the numerical value became our control prototype when designing this survey. We developed five different graph versions in total for users to work with, as shown in Figure 3:

- Graph 1: Darkened arrows. Edge direction is depicted with an arrow, and magnitude is shown by how darkened each arrow is, with a spectrum of shades from light grey to dark grey to black. A magnitude of zero corresponds to a completely white arrow, and arrows corresponding to larger magnitudes progressively darken in color (Figure 3a).

- Graph 2: Tapered lines. Edge direction is depicted by the tip of each thinning line, and magnitude is shown by thickness of the beginning of each line. A magnitude of zero corresponds to a straight line with no tapering thickness (Figure 3b).
- Graph 3: Faded lines. Edge direction is shown by the gradient of light-to-dark fading and magnitude is represented by what percentage of each line is darkly shaded before fading to white. A magnitude of zero corresponds to a completely white line with no shading (Figure 3c).
- Graph 4: Colored arrows. Edge direction is represented with an arrow, and magnitude is shown by both color intensity (from light yellow to dark red) and arrow size. A magnitude of zero corresponds to a yellow, thin line (Figure 3d).
- Graph 5: Numerical values. Edge direction is shown by arrows and magnitude by its numerical value. A magnitude of zero is represented by the number 0.0 (Figure 3e).

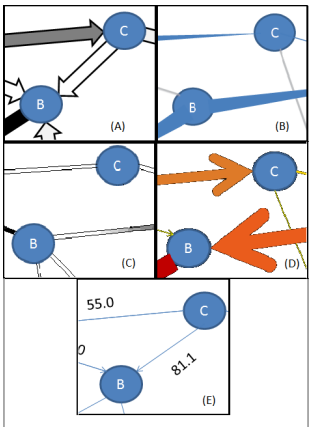

Figure 3: The five graph prototypes we used (from top-left: darkened arrows, tapered lines, faded lines, colored arrows, and numerical values).

To eliminate bias in our graphs, we ensured that all of the graph edge/node pairs were similar, but that none of them were duplicates. We also randomized the order in which each of the graphs appeared for each participant. This was an easy task to accomplish thanks to Bennetts emphasis on Gestalt principles and the emotional design framework that Harary and Norman developed. The Gestalt principles and framework both consist of understanding the ability of a user to derive meaning from shapes, nodes, and edges. Bennetts understanding of aesthetic heuristics supports the foundation for what makes graphs intuitive and easy to understand [14]. One of the main principles of this that our graph prototypes followed is the minimization of the number of edge crossings and the limitation of edge line lengththis is proven to improve graph understandability [15].

III. Hypotheses

From a human-computer interaction perspective, we expected a few qualities about some of the graphs to be intuitive. For instance, we assumed graphs 1, 4, and 5 would have little to no ambiguity regarding graph direction, since direction was denoted with arrows pointing to the target node. It would also follow that graph 5 would have the most clear magnitude, as there arent many different interpretations that one can derive from a numerical value [11]. Our study, however, aims to explore what exactly gives a graph the

property of intuitiveness. Some of the graphs were non-specific in terms of a clear direction, which is intentional on our behalf. We wanted to see if a user would interpret graph 3s directionality as light-to-dark or dark-to-light without additional assistance, or if the user assumed that graph 4s gradient ranged from yellow-to-orange or yellow-to-red without additional assistance. We focused on how users would interpret each graph because of a future concern described in the conclusion of this paperhow much instruction the user should have for any given task. Knowing how to salvage the relationship between the software developer and the user, the motivation for our study, is also a key point our hypotheses try to evaluate.

IV. User Study

After creating the prototypes, we had users perform different graph analysis tasks, including determining the direction of a given edge, the relative magnitude of an edge in comparison to another edge, and if an edge is bidirectional. We also asked users to evaluate the usability of each graph with regards to aesthetics and functionality. The study was online, hosted by Kwik Surveys, and a copy of it is attached in Appendix A [16].

For each prototype, users were asked a series of questions related to the respective graph. The first three questions were related to the magnitude of the values on the edge, the second three questions were related to the direction in which the edges of the graph were pointing. A final question asked the users to self-report on how confident or not-confident they were in answering the questions pertaining to each individual graph. After answering questions about each individual was asked to give each graph a score from one to ten on how easy or difficult they found it to determine direction and magnitude. The final section asked users to answer a series of open ended questions on which graphs they found the easiest and most difficult to use. Participants were selected based on a pool of biological and computational students and faculty at Iowa State University.

V. Results

In order to determine how well the prototypes performed, each graph was analyzed based on three factors. The first factor was user accuracy. Users were asked a set of seven questions for each graph, comprising of three questions regarding edge magnitude, three questions about edge direction, and a final question that prompted each user to evaluate how confident he/she was working with each graph in question (For the complete survey, see Appendix A). The first three questions were grouped together and the percentages of correct answers were taken as the accuracy of magnitude by graph (Figure 4). Errors were broken down into incorrect answers and unable to determine.

A similar process was used with the last three questions to determine accuracy of directionality. As there was no option for unable to answer given, users were forced to choose a direction and the results are given as percent accuracy and percent error.

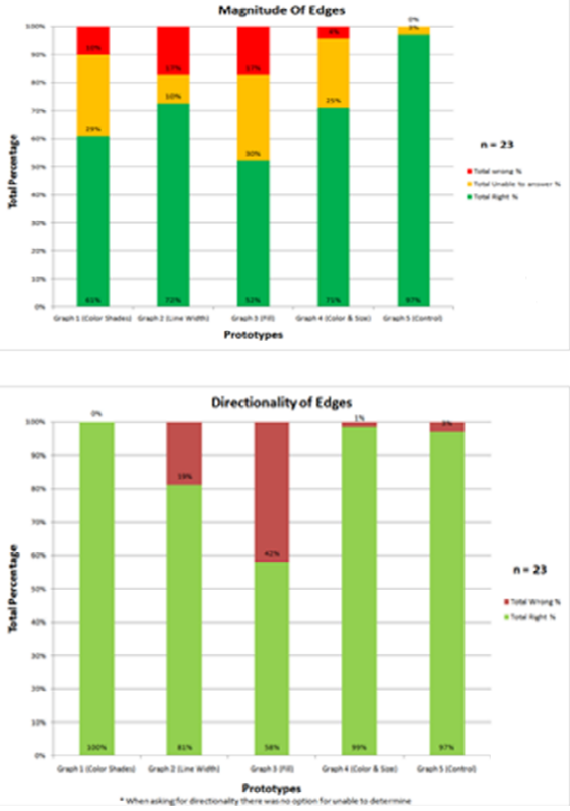

Figure 4: Results from (A) edge magnitude evaluation, and (B) edge direction evaluation, for each graph prototype.

A one-way ANOVA was run in SAS on the data in order to determine the significance of the percent accuracy. For the purposes of the ANOVA, the number of question in each category was used, causing the "unable to determine" choice to be counted as incorrect. Total accuracy, in terms of correct answers, is shown in Figure 5.

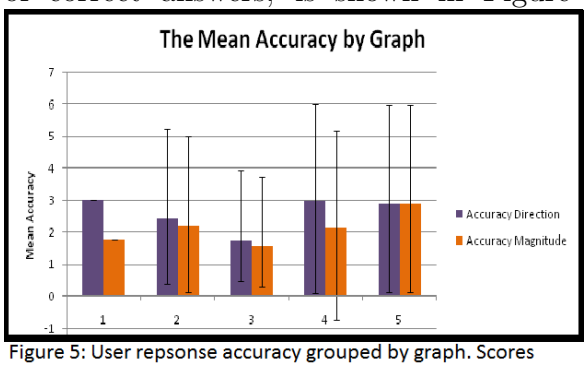

Figure 5: User repsonse accuracy grouped by graph. Scores range from 0 to 3, with 0 being 0% accuracy, and 3 being 100% accuracy.

Immediately after answering the questions about magnitude and direction, users were asked to rank their confidence on a scale of 1 to 5, with 1 being "non-confident" and 5 being "very confident" (the average scores are shown in Figure 6).

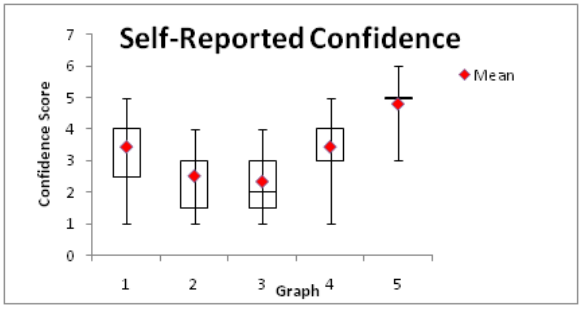

Figure 6: Users rated their confidence in answering questions on the five graphs

A complete list of our results is available in Appendix B.

VI. Discussion

Our survey tested five different prototypes based on two features, directionality and magnitude. The objective of our study was to determine which of our five prototypes performed the best in terms of the user's ability to determine magnitude and directionality. The overall results for magnitude suggests that graph 5 performed the best in displaying magnitude but was not the best when it came to directionality. When analyzing the directionality of the edges, graph 1 performed the best. When testing for directionality, the user was not provided with an "unable to determine", forcing the user to make a selection. This may have impacted the data with an increase in the error percentage, since the user had to guess when left with no other choice. We surveyed a total of 23 participants, of which about 40Based on the accuracy tests, our control, graph 5, had the highest overall score. Graph 4 has a slightly higher score on the directionality (Figure 5). Users reported that they felt the most confident when working with graph 5, which was expected, due to its numerical representations (Figure 6). There is a significant difference in the users ability to determine the magnitude in graph 5 versus all of the other graphs. Outside of the control graph, graphs 1 and 4 had the best scores in both accuracy and direction from the self-reported scores and accuracy tests. From both the self-reported data and the accuracy data there was no real difference between graphs 1, 4, and 5 in the area of directionality. All three of the graphs use a slight variation of a standard arrow notation. Graph 2 had the least amount of difference from the control based on accuracy, closely followed by graph 4. Both graph 2 and graph 4 use a variation of line width to denote magnitude, and the results showed no statistical difference between the two in terms of user response. We also found that Holten's assertion of tapered lines did not show a significantly better user experience [11].

VII. Conclusion/Future Work

The different visualization methods used in this study are not exhaustive and further study may be warranted on different graph types. It may also be beneficial in the future to determine how many different factors should be used to increase user performance. In working with the graphs, we were not able to record the time required for each prototype. This data would help give a more accurate depiction of how difficult or easy a graph is to decipher. It is expected that the control variable would not perform as well on a timed test due to the need to compare each number individually.

Despite the possible modifications to our user study, our findings suggest many improvements for data visualization tools. Both Cytoscape and its flux plug-in FluxViz

prompt users to provide insight to make the programs more usable [5, 10]. Our analysis of these programs and their limitations, presented in this research paper, provides an analysis that

VIII. Acknowledgements

We would like to acknowledge our graduate mentors Erin Boggess, and Jesse Walsh, as well as our faculty mentor, Julie Dickerson, Ph.D. We also thank Iowa State University professors Eve Wurtele, Jacqueline Shanks, post-doc assistant Jong Moon Yoon, and graduate assistant Ting Wei Tee. This study is funded by the National Science Foundation as a part of the Research Experience for Undergraduates program, NSF Grant IIS-0851976. It is hosted by the SPIRE-EIT program at Iowa State University by the Human-Computer Interaction department.

IX. References

1. B. Mirel. Supporting cognition in systems biology analysis: findings on users processes and design implications. Journal of Biomedical Discovery and Collaboration (2009), 4:2, 2009.

2. P. Saraiya, C. North, and K. Duca. Visualizing biological pathways: requirements analysis, systems evaluation and research agenda. Information Visualization (2005), pages 191-205, 2005.

3. P. Saraiya, C. North, and K. Duca. An Insight-Based Methodology for Evaluating Bioinformatics Visualizations, Vol. 11, No. 4 (2005), pages 443-456, 2005.

4. M. Suderman and M. Hallett. Tools for visually exploring biological networks. Bioinformatics (2007), pages 2651-2659, 2007.

5. P. Shannon, A. Markiel, et al. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. Genome Research (2003), pages 2498-2594, 2003.

6. Z. Hu, J. Mellor, J. Wu, and C. DeLisi. VisANT: an online visualization and analysis tool for biological interaction data. BMC Bioinformatics (2004).

7. A. Hagberg, P. Swart, and D. Schult. Exploring network structure, dynamics, and function using NetworkX. In Proceedings of SCIPY (2008).

8. B. Elliott, M. Kirac, et al. Pathcase: pathways database system. Bioinformatics (2008), pages 2526-2533, 2008.

9. C. Krieger et al. MetaCyc: a multiorganism database of metabolic pathways and enzymes. Nucleic Acids Research (2004), pages D438-D442, 2003.

10. M. Konig and H.G. Holzhutter. FluxViz Cytoscape Plug-in for Visualization of Flux Distributions in Networks. Genome Informatics [2010].

11. D. Holten and J. J. vna Wijk. A User Study on Visualizing Directed Edges in Graphs. In Proceedings of the 27th international conference on Human factors in computing systems (CHI 2009), pages 2299-2308, 2009.

12. W. Huang and P. Eades. How People Read Graphs. In Proceedings of the 2005 Asia-Pacific symposium on Information visualization (2005), pages 51-58, 2005.

13. R. Becker, S. Eick, and A. Wilks. Visualizing Network Data. IEEE Transactions on Visualization and Computer Graphics, Vol. 1, No. 1 (1995), pages 16-21, 1995.

14. C. Bennett, J. Ryall, L. Spalteholz, and A. Gooch. The Aesthetics of Graph Visualization. Computational Aesthetics in Graphics, Visualization, and Imaging (2007), pages 1-8, 2007.

15. C. Ware, H. Purchase, L. Colpoys, and M. McGill. Cognitive measurements of graph aesthetics. Information Visualization (2002), pages x-y, 2002.