New Features for GRNsight v2: a web application and service for visualizing models of small- to medium-scale gene regulatory networks

Eileen J. Choe
Loyola Marymount University
Department of Electrical Engineering
& Computer Science
1 LMU Drive
Los Angeles, California 90045
echoe@lion.lmu.edu

Nicole A. Anguiano
Loyola Marymount University
Department of Electrical Engineering
& Computer Science
1 LMU Drive
Los Angeles, California 90045
TBD

Anindita Varshneya
Loyola Marymount University
Department of Biology
1 LMU Drive
Los Angeles, California 90045
TBD

Mihir Samdarshi
Loyola Marymount University
Department of Biology
1 LMU Drive
Los Angeles, California 90045
TBD

Yeon-Soo Shin
Loyola Marymount University
Department of Electrical Engineering
& Computer Science
1 LMU Drive
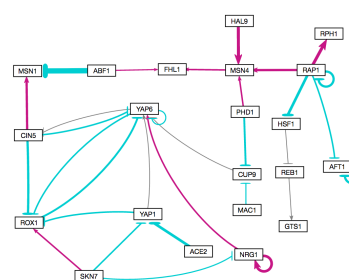Los Angeles, California 90045
TBD

Edward B. Bachoura
Loyola Marymount University
Department of Electrical Engineering
& Computer Science
1 LMU Drive
Los Angeles, California 90045
TBD

John David N. Dionisio
Loyola Marymount University
Department of Electrical Engineering
& Computer Science
1 LMU Drive
Los Angeles, California 90045
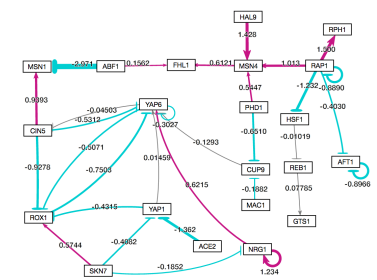dondi@lmu.edu

Kam D. Dahlquist
Loyola Marymount University
Department of Biology
1 LMU Drive
Los Angeles, California 90045
TBD
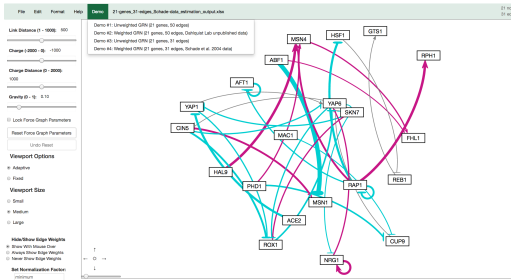
(a) Screenshot of GRNsight   (b) Weighted graph after manual manipulation.   (c) Weighted graph with edge weights displayed.

Figure 1: GRNsight v2's automatic graph layout of Demo #4: Weighted GRN (21 genes, 31 edges, Schade et al. 2004 data) in (a) allows the demo gene regulatory network to fully relax. Zooming and scrolling options allow the user to see the entire graph when it moves beyond the viewport; (b) the user can manipulate the display of the graph through manual node dragging, and can either hide (b), or show (c) all the weight values, which display on the edges.

ABSTRACT

We present new features in v2 of GRNsight, a web application and service for interactive visualization of small- to medium-scale gene regulatory networks (GRNs) [Dahlquist et al. 2016]. A GRN

consists of genes, transcription factors, and the regulatory connections between them which govern the level of expression of mRNA and protein from genes. GRNsight produces weighted or unweighted network graphs from an Excel spreadsheet containing an adjacency matrix where regulators are named in the columns and target genes in the rows, a Simple Interaction Format (SIF) text file, or a GraphML XML file. GRNsight represents genes as nodes

and regulatory connections as edges with colors, end markers, and thicknesses corresponding to the sign and magnitude of activation or repression. For GRNsight v2, the user was given greater control over the network visualization's bounding box and viewport as well as the way edges and their weights are displayed. GRNsight is best-suited for visualizing networks of fewer than 35 nodes and 70 edges, and has general applicability for displaying any small, unweighted or weighted network with directed edges for systems biology or other application domains. The GRNsight application (http://dondi.github.io/GRNsight/) and code (https://github.com/dondi/GRNsight) are available under the open source BSD license.

CCS CONCEPTS

• **Human-centered computing → Scientific visualization**; **Visualization toolkits**;

KEYWORDS

Scientific Visualization, Software Engineering, Bioinformatics

1 INTRODUCTION AND MOTIVATION

GRNsight is a web application and service for the interactive visualization of small- to medium-scale gene regulatory networks (GRNs), optimized for use by novice and experienced biologists alike to quickly and easily view unweighted and weighted network graphs [Dahlquist et al. 2016]. Visual inspection has long been recognized as distinct from other forms of purely numeric, computational, or algorithmic data analysis [Card et al. 1999; Tufte 1986], and GRNsight enables the potential for insight derived specifically by visual inspection. The following are the requirements for GRNsight:

(1) Exist as a web application without the need to download and install specialized software;
(2) Be simple and intuitive to use;
(3) Automatically lay out and display small- to medium-scale, unweighted and weighted, directed network graphs in a way that is familiar to biologists and adds value to the interpretation of the modeling results.

Software Architecture: GRNsight has a service-oriented architecture, consisting of separate server and web client components The server provides a web API that accepts files in Microsoft Excel workbook (.xlsx), SIF (.sif), and GraphML (.graphml) formats and converts them into a unified JSON representation. A converse import function accepts this JSON representation and converts it into either SIF (.sif) or GraphML (.graphml) formats for export. The web application server provides code and resources for the graphical user interface that displays this JSON representation of the graph.

Graph Customizations and User Interface: Graph visualization is facilitated by the Data-Driven Documents JavaScript library [Bostock et al. 2011]. D3.js provides data mapping and layout routines which GRNsight heavily customizes in order to achieve the desired graph visualization. The resulting graph is a Scalable Vector Graphics (SVG) drawing in which D3.js maps gene objects from the JSON representation provided by the web API server onto labeled rectangles. Edge weights are mapped into Bezier curves. The resulting graph is interactive, initially using D3.js's force graph layout algorithm to automatically determine the positions of the gene rectangles. The GRNsight user interface includes a menu/status bar and sliders that adjust D3.js's force graph layout parameters, to refine the automated visualization. Design decisions for the user interface were driven by applicable interaction design guidelines and principles [Nielsen 1994; Norman 2013; Shneiderman 2010] in alignment with the mental model and expectations of the target user base, consisting primarily of biologists.

2 EXTENSIONS TO GRNSIGHT IN V2

Since the release of GRNsight v1, further research, as well as feature requests from peer review have motivated the following improvements to GRNsight v2. We focus on features which have enhanced the visualization and display of the network graph GRNsight produces.

2.1 Separation of Viewport from Graph Bounding Box

The default behavior of D3.js's force graph layout algorithm is to give the graph an a priori bounding box. However, the fixed size did not allow for graphs above a relatively small number of nodes to fully come to rest. Nodes would instead bump up against the edge of the bounding box. This was inefficient, as it resulted in the force graph parameters not being fully applied. To accommodate the range of possible GRNs that GRNsight may display, we revised that algorithm so that the option for an adaptive, expandable bounding box can be chosen. This allows the graph to expand the bounding box as far as required for it to fully come to rest, allowing the force parameters to be fully applied and utilized. To accommodate this change, the ability for the user to zoom and scroll the viewport was also added. Should the user not want to use the adaptive bounding box algorithm, the option to utilize the default fixed behavior is also given.

There are three initial sizes given for both the viewport and bounding box: Small, medium and large. Initially upon loading GRNsight, the best viewport and bounding box sizes are chosen based on the available space on the user's screen. The viewport and bounding box can also be custom-fit to the size of the available space in the browser window. These sizes can later be changed if desired.

2.2 Edge Weight Display Options

A new visualization feature for weighted graphs introduced in GRNsight v2 is an options menu for displaying the edge weights of the graph. Prior to the addition of this options menu, edge weights were only shown upon mouse-over of the edge. When visualizing a graph, the need to be able to see all of the weights of every edge

became apparent. The introduction of this options menu allowed users to choose to always show or always hide weights, as well as the default option of viewing only on mouse-over. Edge Options exist in the sidebar menu, and under the Format tab in the File menu.

2.3 Graph Normalization

To allow for the comparison of weighted network graphs, a feature request was made for the user to have the option to customize the normalization factor applied to the edge thickness of the graph. By default, GRNsight detects the maximum and minimum relationship values of a network, and normalizes the data to fit within 12 distinct edge thicknesses corresponding to the strength of its regulatory relationship. However, with this model, a graph with weights in the range -1 to 1 could appear the same as a graph with weights in the range -10 to 10. By adding the option to user-specified minimum and maximum values, the edge thicknesses can be consistently normalized in a weighted graph, so the user can compare different graphs on the same scale.

(a) Network A

(b) Network B without Normalization   (c) Network B with Normalization
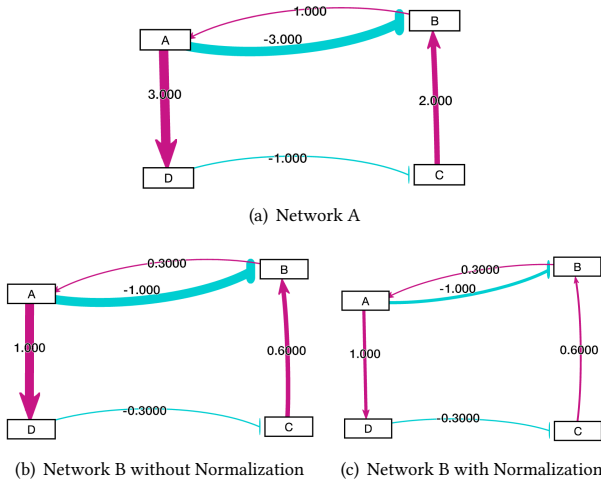
Figure 2: GRNsight automatically detects the minimum and maximum weights in (a) and (b) to calculate the edge weights. By overriding the maximum and minimum values of Network B to -3, 3 in (c), Network B is now normalized to match the scale of Network A. Now the networks in (a) and (c) can be compared.

3 CONCLUSION AND FUTURE WORK

We have successfully implemented GRNsight, a web application and service for visualizing small- to medium-scale GRNs that is simple and intuitive to use. GRNsight reads a weighted or unweighted representation of a GRN, and automatically lays out and displays unweighted and weighted network graphs, enabling interpretation of the weight parameters more easily than one could from an adjacency matrix alone. Extensions to GRNsight in v2 have expanded GRNsight's visualization and layout capabilities, giving the user more control over the visual display of the network graph.

GRNsight is in active development, we would like to further enhance the tool to compute and display graph statistics, as betweenness centrality measure. Further research will be conducted to evaluate different graph layout options, such as a hierarchical or block layout and the effectiveness that alternate visualization paradigms for the biologist...

REFERENCES

M. Bostock, V. Ogievetsky, and J. Heer. 2011. D3; Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (Dec 2011), 2301–2309. https://doi.org/10.1109/TVCG.2011.185

Stuart K Card, Jock D Mackinlay, and Ben Shneiderman. 1999. *Readings in information visualization: using vision to think.* Morgan Kaufmann.

Kam D Dahlquist, John David N Dionisio, Ben G Fitzpatrick, Nicole A Anguiano, Anindita Varshneya, Britain J Southwick, and Mihir Samdarshi. 2016. GRNsight: a web application and service for visualizing models of small- to medium-scale gene regulatory networks. (Sep 2016). https://peerj.com/articles/cs-85/

Jakob Nielsen. 1994. *Usability engineering.* Elsevier.

Donald A Norman. 2013. *The design of everyday things: Revised and expanded edition.* Basic books.

Ben Shneiderman. 2010. *Designing the user interface: strategies for effective human-computer interaction.* Pearson Education India.

Edward R. Tufte. 1986. *The Visual Display of Quantitative Information.* Graphics Press, Cheshire, CT, USA.