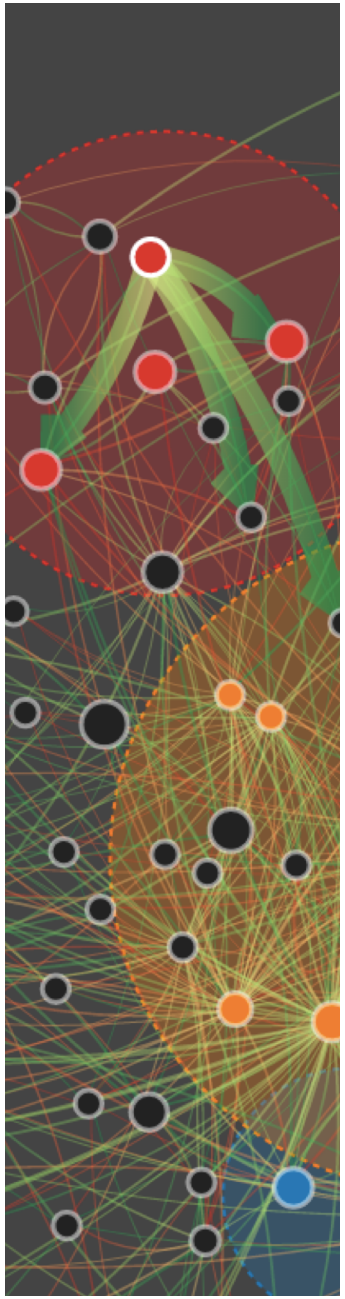

Interactive Visualization of Dynamic Influence Maps Through Clustered Force-Directed Networks

Andrew Burks
aburks3@uic.edu

Kristine Lee
khlee2@uic.edu

Xing Li
xli227@uic.edu

CONF '22, Jan 1 - Dec 31 2022, Authorberg.
 ACM 978-1-xxxx-yyyy-z/zz/zz…$zz.00.
unique doi string will go here

Abstract

Force-directed networks have often been used to generate aesthetically pleasing visualizations of various relationships. However, there are numerous issues whe with using force-directed networks such as scaling, visual clutter and abstracting meaningful information in large networks. In this paper, we apply force-directed networks to a biological system and explore various techniques that can be used to help alleviate the issues faced in larger force-directed networks.

Introduction

In biology, many times, the parts that make the biological system run are known. However, for many systems, the specifics regarding the interactions and effects within the system are unknown. Kappa Simulator (KaSim) is a rule-based language and stochastic simulator. It is used, in this case, for simulating protein interaction networks. A tool to represent the results of these simulations can help the scientists better understand the stochastic biological systems they are modeling. If successful, this creates a feedback loop between the creation and the visualization of the simulations which iteratively improves both the system models as well as the visual analytics tools which aid in the understanding of the

biological systems. The simulation results produced follow a standard format. However, a visual analytics tool is required to help analyze and understand these systems. The current tools do not scale well, are static, cluttered, and cannot support time-series visualization.

Topological analyses can be very useful in understanding interactions within networks. However, as link density increases, the readability of node-link diagrams decreases. While other techniques can efficiently scale to networks of higher sizes and link density, it can be difficult to draw insight from these visual representations as they have a higher learning curve. Furthermore, when using a node-link diagram, network layout is an optimization problem. In many cases, finding an ideal arrangement can be a non-converging problem. It is necessary when visualizing this data to create a tool which can scale well while remaining a rich source of information regarding the simulation. This work presents an open-source, web-based tool to visualize Dynamic Influence Maps for protein interaction networks simulated through KaSim. Our tool includes a clustered force-directed network. User interaction allows manual network arrangement, data filtering and control over the real-time, single-parameter clustering.

Related Works

There are several proposed solutions for conveying dynamic visualization maps, ranging from matrices to chord diagrams to node-link graphs. Our motivation for using a node-link graph was that it was particularly intuitive for the given domain, especially for non-visualization experts. Some of node-link graphs' common problems that have been touched upon in other research: namely the problem of an optimal layout and difficulties in capturing a sense of time from a graph.

To alleviate the issue of overlapping nodes, we used a force directed simulation. For dense hairballs of edges, we adopted the pipeline proposed in [1]: when transforming the data into a graph, taking the additional steps of filtering and clustering could improve the layout of dense graphs. This filtering technique centers on identifying "seed nodes," or important nodes with high degrees of connectivity, and "seek nodes," which satisfy a certain threshold, to generate clusters.

For time series visualizations of node-link graphs, there are usually two options: to animate the graph through each time step to its optimal layout, or to anchor the positions of nodes, but potentially result in more edge crossings [2]. Because of the nature of the data we are handling, we felt that anchoring the rules in place during time-based animations gave a better sense of how the graph changed over time.

Analytic Tasks

Output from KaSim simulations contained the following information: the *time window*, which measured when and for how long the data were measured; the *rule*, which identified the sites being activated and manner of interaction; the number of *hits*, or number of times that the rule was invoked during the time window; and a matrix of *influences*, which measured the degree to which the activation of one rule affected the activation of another rule. Extended simulations could potentially generate dozens or hundreds of files.

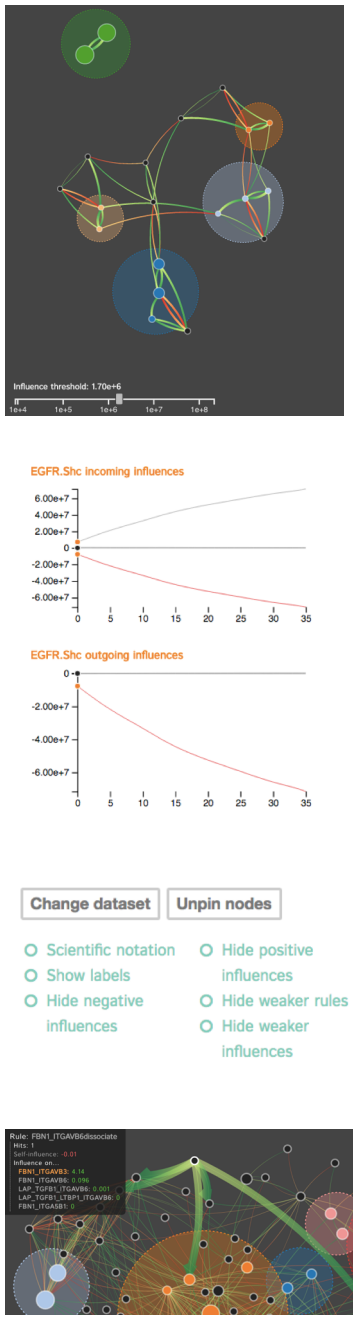

Figure 1. Interactive Elements: top- clustering, second – supplemental vis, third – buttons, bottom - hovering

For a dynamic influence map, the primary analytical task was tracking changes in influence over time. Because of the amount of information encoded with each rule at every time step, it was necessary to filter out the most important relationships at a glance, while also providing the ability to bring up more detailed information on demand. Additionally, there needed to be a way to communicate data changes over time, while maintaining continuity throughout the entire time.

Visualization Details

Using D3.js, we developed an interactive web tool that overlays a force-directed network on top of influence-based clustering information for the rules. To address the needs of in-depth and overview information, we divided the interface into two interactive visualization components: the network layout of the entire model at a specific time step and a line graph charting the changes in a single rule over the span of the entire simulation.

When factoring in the expectation that protein interaction simulations tend model a low number of rules (fewer than 100), the node-link graph is an intuitive visualization for grasping the relationships and degrees of connectedness between rules. The layout of closer together are connected by strong influences; second, negative influences repel while positive influences attract. Thus, a general impression of the relationships between rules can be gleaned from layout alone. Although it makes semantic sense to group together the rules that positively influencing each other, the former took priority in order to reduce visual clutter.

Other visual encodings included using size of the circle to convey the number of hits on a rule, a gradient to indicate the direction of the acting influence, color to categorize influences as positive or negative, and line thickness to convey the magnitude of influence.

Additionally, we adopted a simple, single-parameter clustering method whose threshold can be controlled manually. To reduce clutter, the user may choose to manually rearrange areas of the network, as well as hide nodes or edges which fall below the clustering threshold.

Network layouts rapidly become difficult to understand as the number of nodes and degree of connectedness increase. With the range of data sizes we expected, the network layout can still be managed with some filtering.

There are several ways of interacting with the data. First, the clustering threshold mentioned above can be used to manually cluster influences above a certain threshold. Second, the toolbar contains various options that can be used in conjunction with the clustering threshold to filter out rules and influence paths that are above the threshold or filter based on positive or negative influences. Furthermore, by hovering other a rule, all outgoing influences will be highlighted while all other influences with be dimmed and a supplemental visualization showing outgoing and incoming influences of the rule over time will be displayed on the lower right of the application. Hovering over rules and influences with also provide a tooltip at the upper-right-hand corner which will display more information

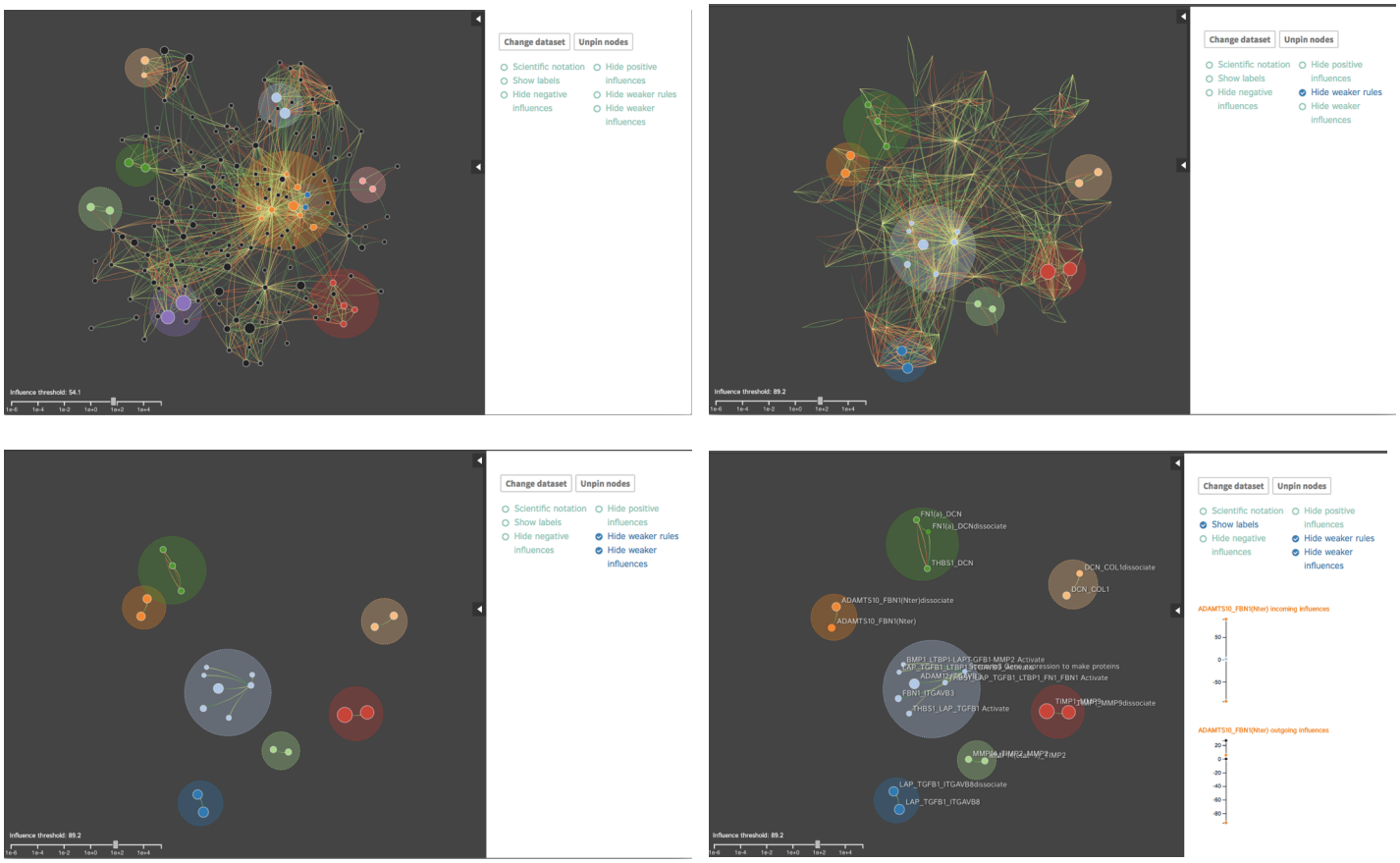

Figure 2. a) upper-left: Large dataset with no filtering b) upper-right: Large dataset with rules below threshold removed

c) lower-left: Large dataset with rules and influences below threshold removed d) lower-right: graph with labels

regarding the rule/influence. Finally, dragging rules or cluster circles allows the user to manually control the layout where the force directed simulation failed to perfect it.

These options can be used while moving through the time series data to discover interesting rules and patterns over time while reducing the clutter that is often present in force directed networks. These interactions are demonstrated in Figure 1.

Case Study

One of the major user requirements was to be able analyze time series data from KaSim outputs. The data was in a specified json format where each rule was labeled with its name and hits along with a matrix of its influences to other rules. The time series data was generated over a moving sliding window (10s) so each time step represented the average values over that period.

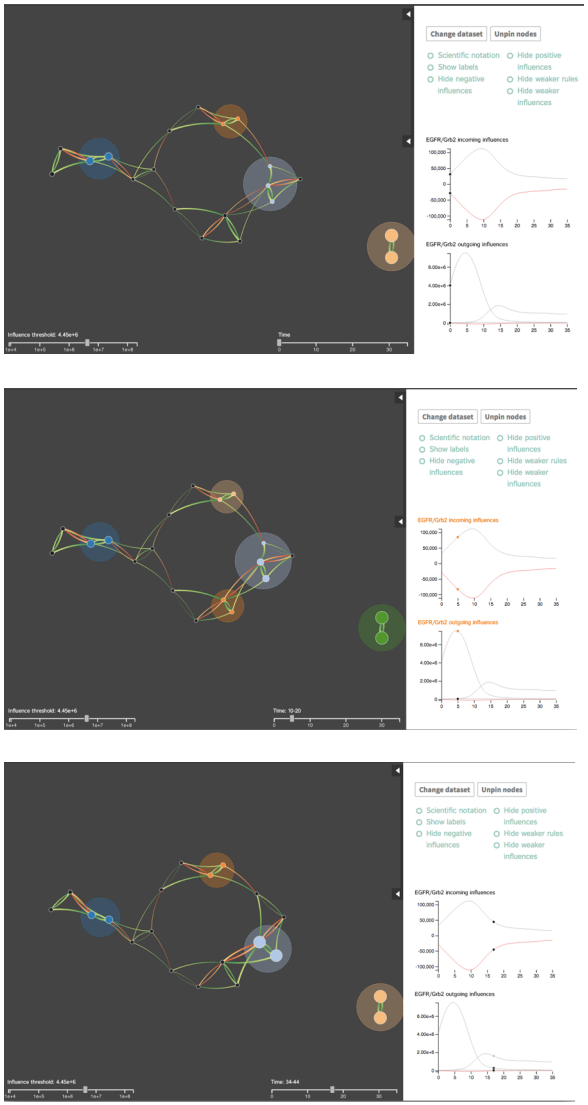

Figure 3. Sample Time Series Data

We used our visualization to analyze sample data generated from KaSim, courtesy of Fontana Labs. Two kinds of datasets were tested: a time series dataset with 35 time windows; a large set of over 160 rules (Figure 2a-b). Although the network layout alone does look clustered at the start, once the dataset was filtered, the layout become a lot more manageable (Figure 2c-d). The time series data highlights changes in rules (and by extension the clusters) and influences themselves as can be seen in Figure 3.

From the figures, one can see that there are interesting changes happening in the time series data, hopefully being able to see these changes will allow domain experts to narrow down the specific pathway that rules affect one another and itself.

Expert Feedback

When the case study was presented to domain experts, the visualization was received positively. In particular, the ability to see an overview of all the rules and interactions in one graph and clustering of nodes in real time were praised. Because of the difficulty in making sense of KaSim datasets in previous efforts, experts were interested in exploring the tool to first better understand what information they could glean from their data at all, and second, to identify what features needed focus on in the future. Further suggested developments include highlighting significant "chains of influence" between connected rules.

Conclusion and Future Work

KaSim provides a way of extracting potentially interesting information about complex biological networks, but so far, efforts to visualize data generated by the simulator have left much to be desired. The proposed visualization in this article organizes and lays out data to highlight significant relationships in a way that is both intuitive and descriptive.

On a higher level, an effective visualization for modeling dynamic influence maps of rule-based systems in general could benefit for any dynamic system that is described with rules and influences.

Acknowledgements

Thank you to our instructors in the project, Dr. Angus Forbes & Shiwangi Singh, as well as our collaborators at Fontana labs, Dr. Walter Fontana and Pierre Boutillier, who provided the datasets and explanations of the domain.

References

1. Xiaodi Huang, Peter Eades, and Wei Lai. 2005. A framework of filtering, clustering and dynamic layout graphs for visualization. In Proceedings of the Twenty-eighth Australasian conference on Computer Science - Volume 38 (ACSC '05), Vladimir Estivill-Castro (Ed.), Vol. 38. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 87-96.ACM. How to Classify Works Using ACM's Computing Classification System. 2014. Retrieved August 22, 2014 from http://www.acm.org/class/how_to_use.html.

2. James Moody, Daniel Mcfarland, and Skye Bender-Demoll. 2005. Dynamic Network Visualization. American Journal of Sociology 110, 4 (2005), 1206–1241. DOI:http://dx.doi.org/10.1086/421509