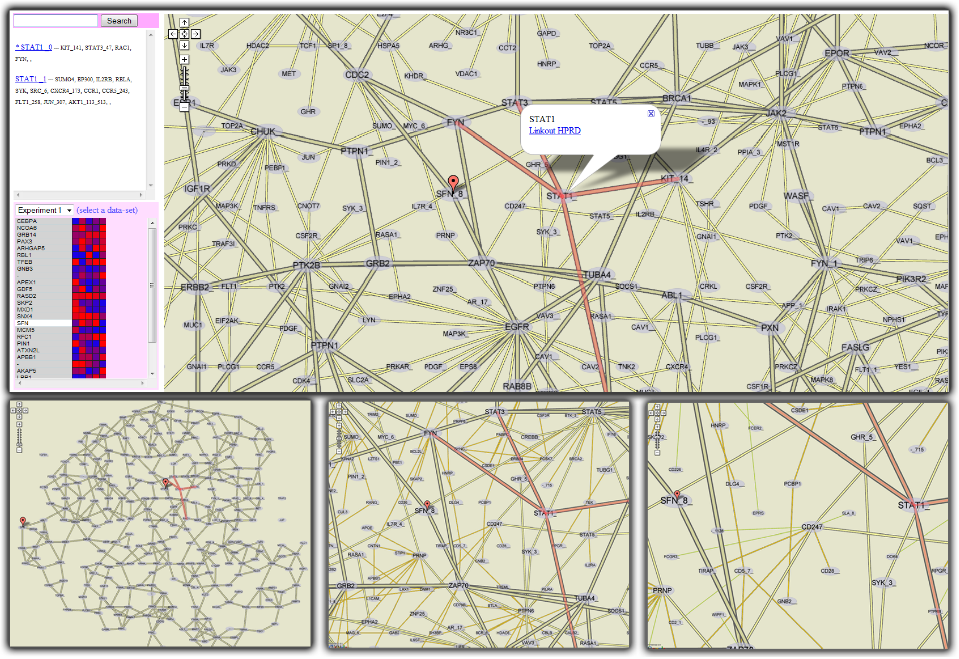
Poster: Visualizing Protein Interaction Networks as Google Maps

Radu Jianu[*] David H. Laidlaw[†]

Brown University



Search

* STAT1 _0 — KIT_141, STAT3_47, RAC1, FYN, ,

STAT1 _1 — SUMO4, EP300, IL2RB, RELA, SYK, SRC_6, CXCR4_173, CCR1, CCR5_243, FLT1_258, JUN_307, AKT1_113_513, ,

Experiment 1 (select a data-set)

CEBPA
NCOA6
GRB14
PAX3
ARHGAP5
RBL1
TFEB
GNB3
-
APEX1
GDF5
RASD2
SKP2
MXD1
SNX4
SFN
MCM5
RFC1
PIN1
ATXN2L
APBB1
-
AKAP5

STAT1
Linkout HPRD

Figure 1: Protein interaction network and quantitative proteomic data. The top panel shows an overview of the analysis setup: time-course proteomic data is displayed on the lower left and the protein selected in the list is highlighted on the map. A second protein was selected on the map and has its interactors and meta-information displayed. All instances of this protein are listed on the upper left, with their interactors. Additional zoom levels are shown at the bottom: as zoom level increases, less relevant proteins are added to the display.

Abstract

We present a lightweight approach for exploring large protein interaction networks in browsers using the Google Maps API. Feedback from an anecdotal evaluation shows that immediate access to data, low learning overhead and familiarity are appreciated by proteomic researchers, that linking map-zooming to a protein-relevance filter to create a multi-level layout of interacting proteins is desirable, and that, in the context of protein interaction networks, protein duplication is an acceptable technique that can keep interacting proteins close together while removing clutter. Finally, we show how digital map features and additional client graphics can support analysis of experimental data in the context of known interactions.

Keywords: Network Data, Bioninformatics, Presentation and Dissemination

1 Introduction

Understanding how proteins interact to generate cellular responses to external events could enable researchers to alter a cell's behavior without introducing side effects. We describe an accessible approach with low learning overhead: static drawings of large protein interaction networks are served using the Google Maps API with a few familiar interactions (*www.cs.brown.edu/people/jr/ProteinNetwork/network.html*). An evaluation shows that this approach can meet researchers' basic exploration needs and validates our design choices.

Systems for visualizing protein interaction networks already exist [8, 1] but assume small network sizes and interactivity. Alternatively, we offer quick access to data without installation and visualization-creation overhead by exploring the possibility of representing large networks statically while keeping the visualization usable.

Web distributed visualization is not new. Even proteomic systems have been made available as applets or web-client applications [8]. However, users still need to control parameters when creating visualizations, specify data queries, and learn features, all of which constitute undesired overheads. Close to our work is the use of Ajax (asynchronous JavaScript and XML) to render images on the server and asynchronously send them to a client browser. A call for Ajax-based applications in biology and exemples of such an approach have recently emerged [3, 5, 7]. However, the sole difference from offline visualization systems is that control and display happens in a separate place from rendering and computation. Our research differs by limiting regular users' effort in creating visual-

[*] e-mail: jr@cs.brown.edu
[†] e-mail:dhl@cs.brown.edu

izations and assigning this task to experienced personnel, and by using the Google Maps API, a readily available Ajax framework. Finally, work that uses Google Maps to visualize genome structure has become available [10, 2]. We extend this work to protein interaction networks and perform an evaluation that reveals design insights.

2 Methods

We use vertex splitting and filter through zooming to overcome two challenges when displaying static networks: clutter and the fact that zooming does not guarantee a useful data query because related data is not necessarily co-located (e.g. for long edges).

To ensure co-location of linked proteins we use vertex splitting [6] with an optimized force-directed layout [4]. Once the network's spring system reaches stability, tensions on nodes determine the need for a node split. Given a dividing line through a node, force vectors are added on each side of the line. Multiple divisions are probed to find a node's maximum tension. The node with maximum tension is split if the tension exceeds a threshold: two copies of the node are created and edges are assigned depending on whether the force vectors they created were on one side of the split or the other.

To deal with clutter, we link zoom to a protein-relevance filter. A protein's relevance is a combination of intrinsic relevance and relevance diffused from neighboring proteins [9] in a way that avoids elevating relevance for proteins connected only to one highly relevant protein, such as in satellites of protein hubs. A protein's intrinsic relevance is a mix of protein degree and occurrence in pathways or experimental data-sets of interest to our collaborators. The relevance score is used to partition proteins in bins corresponding to zoom levels, much like the city-versus-town distinction in a map analogy. At rendering, nodes are displayed only if their relevance-bin index is lower than a threshold based on zoom level. Node sizes reflect differences in relevance while preserving a sense of uniform scale throughout zoom levels.

The layout is performed in stages, one for each bin. The most relevant proteins are laid out first, and their positions then frozen for the second bin to be placed on the map. Because current-bin layouts are not aware of future-bin graph topologies, layouts can be suboptimal. Thus, we allow two or more bins to be laid out concurrently while "prioritizing" the current one - network elements in non-current levels provide guidance for the current layout by exerting less force than those in the current level.

Ultimately, the visualization is rendered to Google-Maps tiles. To facilitate selection, we export for each tile a text file containing protein bounds, or parts of proteins that appear in the tile. Upon mouse-clicks, the target tile's content file is retrieved and used to check for intersections. This implementation conforms to the data-on-demand architecture and avoids the need to load large data files. As shown in Fig. 1, we use polyline overlays to highlight a selected node's neighbors, information pop-ups to display meta-data, and markers to highlight experimentally derived proteins. To navigate between copies of the same protein, we list all copies of a protein and their interacting proteins such that clicking on copies causes the map to pan to the specific location.

Finally, time-course experimental data can be loaded and displayed as heatmaps on the left-hand side of the map. Multiple experimental datasets can be loaded and toggled between during analysis. Upon an experimental protein selection, markers will indicate the map location of the protein.

3 Evaluation

We evaluated our setup with four proteomic researchers from two different labs. Our subjects were excited about looking at interaction networks in their browsers. The consensus was that the browser setup is effective and that they would choose it over other systems they were familiar with. They explained that they don't like to spend time installing software and learning new features, and found the techniques we demonstrated intuitive and easy to use.

Feedback was also positive on the design decisions underlying the map visualization. The unanimous opinion was that relevance filtering was intuitive and that it corresponds to the normal analysis of a new network: identify important or familiar proteins and then learn more about their neighbors. Another comment was that seeing familiar proteins and connections early reinforces their confidence in the visualization. None of our subjects thought that not seeing the whole network at once obstructed their exploration, while one explicitly stated that the simplified view is superior to cluttered network visualizations he has seen before. All of our subjects were satisfied with how protein relevance was computed.

The first researcher we interviewed found the concept of split proteins disorienting when the concept was described to him but revised his position once we demonstrated the hyperlinked copies list (Figure 1). This allowed him to go through the copies of a protein systematically to reconstruct its neighborhood, and he stated that while splitting proteins is not desirable, it is acceptable if it can simplify the visualization. Our other three subjects said that multiple copies of proteins would not get in the way of their analysis at all; one even said he preferred looking at proteins this way because it made their interaction neighborhoods more apparent. Another subject noted that pathway drawings often had multiple copies of proteins. When we pointed out that these copies are biologically motivated while ours aren't he agreed but said they are still familiar with the technique. Generally, it seemed that the hyperlinked copy list enables users to perform their primary task: finding all interactors of a protein.

4 Conclusion

We introduce the paradigm of visualizing protein interaction networks as digital-maps using the Google Maps environment. We present design choices that make this concept possible while working well in this domain. We present an anecdotal evaluation that validates our design choices and provides additional insight into the use of protein interaction visualization.

References

[1] Ingenuity. `http://www.ingenuity.com/`.

[2] K. Arakawa, S. Tamaki, N. Kono, N. Kido, K. Ikegami, R. Ogawa, and M. Tomita. Genome Projector: zoomable genome map with multiple views. *BMC bioinformatics*, 10(1):31, 2009.

[3] G. Aravindhan, G. Kumar, R. Kumar, and K. Subha. AJAX Interface: A Breakthrough in Bioinformatics Web Applications.

[4] J. Barnes and P. Hut. A hierarchical 0 (N log iV) force-calculation algorithm. *nature*, 324:4, 1986.

[5] S. Berger, R. Iyengar, and A. Ma'ayan. AVIS: AJAX viewer of interactive signaling networks. *Bioinformatics*, 23(20):2803, 2007.

[6] P. Eades and C. De Mendonca. Vertex splitting and tension-free layout. *Lecture Notes in Computer Science*, pages 202–211, 1995.

[7] B. Gretarsson, S. Bostandjiev, J. ODonovan, and T. H
"ollerer. WiGis: A Framework for Scalable Web-based Interactive Graph Visualizations.

[8] P. Shannon, A. Markiel, O. Ozier, N. Baliga, J. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 13(11):2498, 2003.

[9] F. van Ham and A. Perer. Search, Show Context, Expand on Demand: Supporting Large Graph Exploration with Degree-of-Interest. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):953–960, 2009.

[10] T. Yates, M. Okoniewski, and C. Miller. X: Map: annotation and visualization of genome structure for Affymetrix exon array analysis. *Nucleic Acids Research*, 36(Database issue):D780, 2008.