Applying Information Visualization Principles to Biological Network Displays

Tamara Munzner

University of British Columbia

ABSTRACT

We use the principles of information visualization to guide the design of systems to best meet the needs of specific targets group of users, namely biologists who have different tasks involving the visual exploration of biological networks. For many biologists who explore networks of interacting proteins and genes, the topological structure of these node-link graphs is only one part of the story. The Cerebral system supports graph layout in a style inspired by hand-drawn pathway diagrams, where location of the proteins within the cell constrains the location within the drawing, and functional groups of proteins are visually apparent as clusters. It also supports exploration of expression data using linked views, to show these multiple attributes at each node in the graph. The Pathline system attacks the problem of visually encoding the biologically interesting relationships between multiple pathways, multiple genes, and multiple species. We propose new methods based on the principle that perception of spatial position is the most accurate visual channel for all data types. The curvemap view is an alternative to heatmaps, and linearized pathways support the comparison of quantitative display as a primary task while showing topological information at a secondary level.

1. INTRODUCTION

Information visualization systems provide visual representations of datasets intended to help people carry out some task more effectively. In short, pictures help us think by substituting perception for cognition. The value of using external representations is that it frees up the limited amount of cognitive and memory resources for use in higher-level problems.[1] For example, when confronted with a table of numbers that represents measured gene activity, understanding the overall pattern of the data requires considerable conscious thought and attention: numbers must be read, internally stored, and internally compared against other numbers to find values like the min and the max. It is difficult to store enough information in our limited mental buffers to have a detailed overview of the numerical distribution. In contrast, when we look at a visual encoding of the information using a visual channel like color to show those values, we can immediately see overall patterns without conscious effort.

Visualization is appropriate in when two conditions are met. First, when there is a need for the human to be in the loop and the goal is to augment, not replace, human cognition. It is relevant for the many problems that cannot be automated. For instance, the problem may be sprawling and not fully understood, with complex and shifting decision criteria. Another common use of visualization is when a proposed automatic solution must be vetted by people before deployment. Second, when simple summary statistics alone are not adequate. While statistical measures are a powerful and necessary tool for characterizing a dataset, they often do not adequately characterize the full story of the complex distribution of values within the dataset.

Anscombe's quartet is a compelling example.[2] It is four datasets with identical statistical properties, including mean, variance, correlation coefficient, linear regression lines, and several other measures. However, looking at a visual representation of the datasets in the form of scatterplots instantly shows the completely different characteristics of the datasets, as shown in Figure 1. One is a loose linear distribution, another is clearly nonlinear, one is a very tight linear distribution with one outlier, and one is linear in a completely different direction from the regression line with a very different single outlier. Although this dataset is quite small, the idea carries through to larger and more complex datasets: we often need to see the details that underlie simple summaries.

Visualization can facilitate not only hypothesis discovery, but also hypothesis confirmation – and disconfirmation. Data cleansing or wrangling is just one example; the nearly inevitable result of loading even the most well-curated dataset into a new visualization system is discovery of previously unknown problems. Despite the

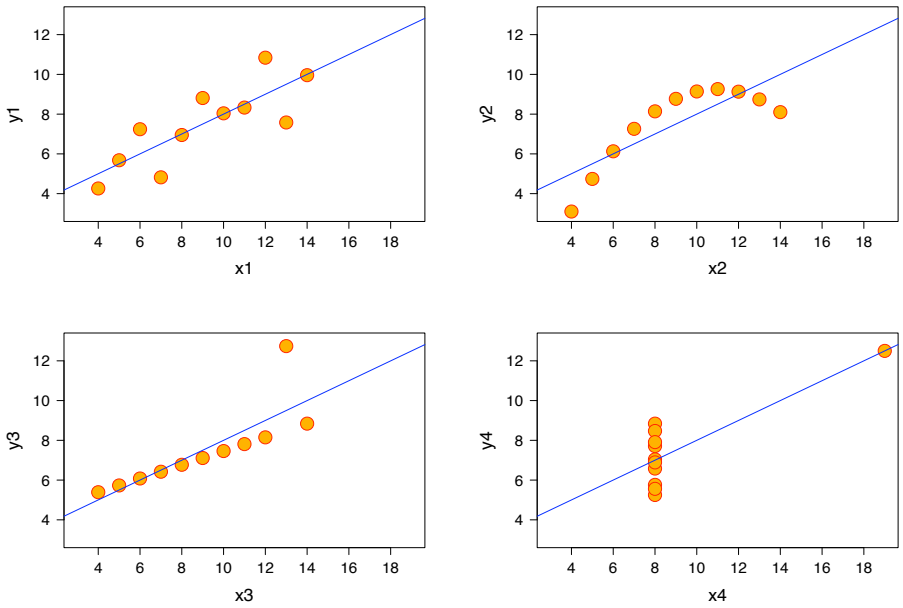

Figure 1. Anscombe's Quartet shows four datasets with identical simple statistical properties: mean, variance, correlation, and linear regression line.[2] However, visual inspection immediately shows how their structures are quite different. Image from `http://en.wikipedia.org/wiki/File:Anscombe's_quartet_3.svg`, under CC Attribution-ShareAlike license.

frequent mindset that the main goal of visualization is to provide fundamentally new capabilities, the most common case is that visualization provides speedup by accelerating existing workflows, as Christian Chabot of Tableau pointed out in his VAST 2008 keynote.

2. SEPARATION OF CONCERNS

Designing an effective visualization is a tricky task, because there is such a huge space of design alternatives and so many of the choices lead ineffective systems. Principled reasons to make these choices are usually not obvious to untrained people, and moreover there are often conflicting tradeoffs that make the design a difficult problem even to people who have visualization design experience. The nested model of visualization design provides guidance in designing, analyzing, and validating visualization systems by separating the set of concerns into four different levels, with different threats to validity at each level.[3]

The nested model has four levels: problem, abstraction, encoding and interaction, and algorithm. The first level is characterizing the problems of real-world users: identifying gaps, breakdowns, and slowdowns that might be addressable by a visualization system. The threat to validity at this level is "users don't do that". Understanding the problems of real-world users is a difficult problem that can be attacked through methodologies from human-computer interaction and ethnography.

The second level is abstracting from domain problems into operations on data types. Generic operations include sorting, filtering, browsing, comparison, characterizing trends and distributions, finding anomalies and outliers, finding correlation, and so on. Data types include tables of numbers, relational networks, and spatial fields. Often the abstraction involves extensive transformations to the original data into a more useful configuration, in order that the operation can be more effectively addressed by drawing the derived data. The threat to validity at this level is "you're showing them the wrong thing". Identifying the set of operations and data types that are used as the building blocks of visualization is a fundamental part of the information visualization research agenda.

Designing visual encoding and interaction is often considered the heart of visualization. We can analyze visual encodings in terms of two crosscutting aspects: geometric marks such as points, lines, and areas; and visual attributes that communicate information atop these marks using channels such as spatial position, color,

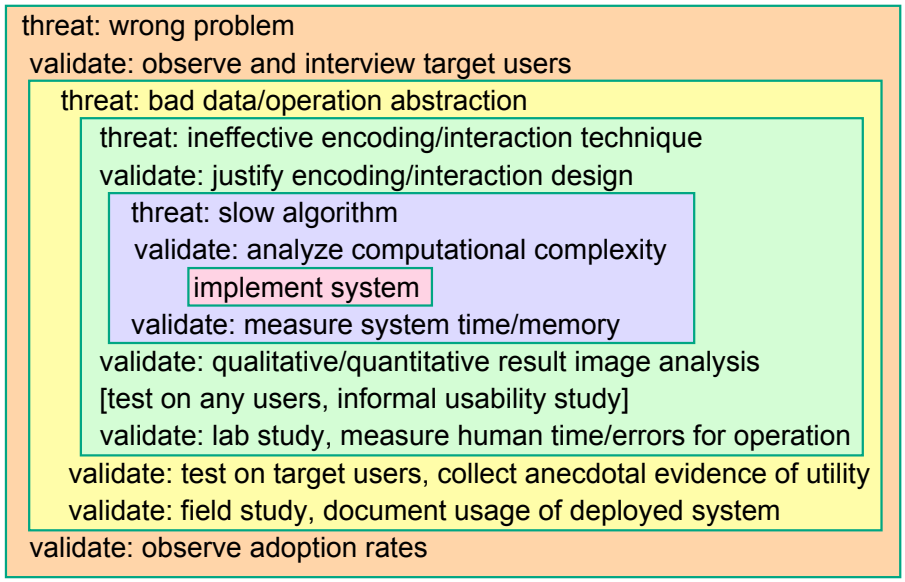

Figure 2. The nested model proposes a split into four different levels of visualization design choices, with different validation methods at each level based on the relevant threats.[3]

shape, size, orientation, and so on. Interaction choices include selection, navigation, reordering, and many other ways to change the visual representation dynamically. The threat to validity at this level is "the way you show it doesn't work". Weighing the effectiveness of encoding and interaction techniques requires knowledge of human perception, drawing on cognitive psychology and vision science.

The fourth level is creating algorithms to automatically execute the techniques specified at the level above. The threat at this level is "your code is too slow". Algorithm design and evaluation are classic computer science problems, and sometimes this layer of design benefits from computer graphics and computational geometry knowledge.

The benefit of separating the design into these four levels is that we can use appropriate methods for validation at each of the levels, in order to understand how and whether the visualization system has achieved its goals. Figure 2 summarizes the framework. At the problem level, an immediate validation approach is to interview and observe the target users to ensure that the problem is correctly understood by the designers. One downstream way to validate whether a visualization system has solved its intended problem is to note the extent to which is is adopted by members of the target user group. Adoption is of course a noisy signal, in that many good systems do not win in the marketplace and vice versa. Nevertheless, knowing that a system has crossed the gap from a research prototype used only by the authors of the system to a system in active use does provide some useful information.

A good approach to validation for the abstraction level is to observe usage by the target group in real-world contexts with their own data. This observation may take the form of formal and systematic field studies, or more informal deployment to collect anecdotal evidence of whether the users are in fact able to work faster or more effectively. At the encoding and interaction level, an immediate form of validation is to carefully justify the design according to known principles. One downstream validation approach is the qualitative or quantitative analysis of the visualization system output, for example by comparing images generated by one visualization system to those from another. Another is a formal measured-response laboratory experiment, to measure human time and errors in carrying out abstract tasks using the system. Finally, the algorithm level can be validated immediately with computational complexity analysis, and downstream by running benchmarks to measure system time and memory usage.

We developed the nested model in part from the experience of designing visualization systems in a wide range of domains, including computer networking,[4] computational linguistics,[5] web logs,[6,7] large-scale system administration,[8] and several areas of biology.[9–11] We will illustrate the role of these information visualization

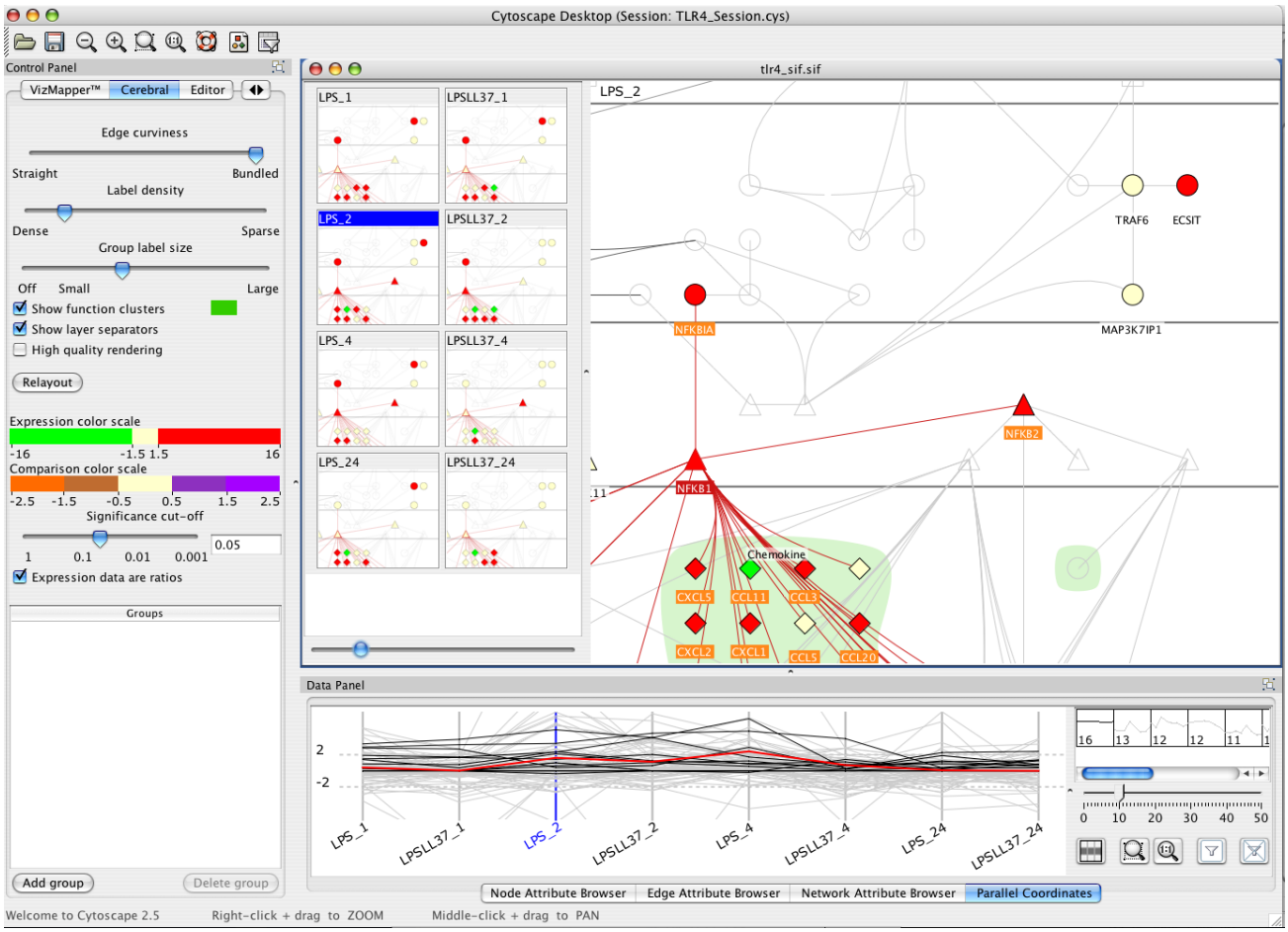

Figure 3. The Cerebral interface includes a large graph interaction window, small multiples colored by experimental condition, and a parallel coordinates view of the measurement data.

principles with a detailed walkthrough of the design decisions of two visualization systems that deal with biological networks, Cerebral[12] and Pathline.[13]

3. CEREBRAL

The Cerebral system [12] was joint work with Aaron Barsky, Jen Gardy, and Robert Kincaid. It was designed in collaboration with innate immunity researchers at the UBC Hancock lab who needed to compare gene activity levels from multiple experimental conditions within the context of a systems biology graph model. Figure 3 shows the Cerebral interface, which includes a large window for interacting with the systems biology graph, small multiple windows with the graph nodes colored by experimental condition, and a parallel coordinates view on the bottom showing the experimental measurements directly.

We will discuss design choices at three of the levels, starting the problem of supporting the model-experiment cycle in systems biology. We will then cover the visual encoding choice of creating a custom graph layout and the algorithm choice of using simulated annealing for layout. We will then discuss two more choices that are at the encoding and interaction level: using small multiple views of the graph, and showing side by side views of both the graph and the measured data. We will conclude with the Cerebral example in a discussion of validation.

3.1 Problem: systems biology

A pervasive model in systems biology is the network of known interactions between biomolecules. In this node-link graph, nodes represent proteins, genes, DNA, or RNA; links represent known interactions. These graphs

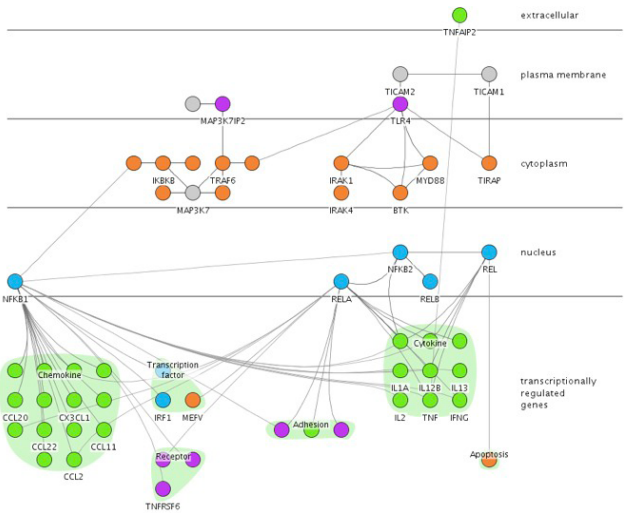

Figure 4. Local neighborhood around the TLR4 biomolecule, with 54 nodes and 74 edges.

are carefully curated, where each link has provenance information about the specific publications that provide evidence for its existence. They are not only complex, but also constantly updated as research progresses, so even biologists very familiar with a specific field do find value in seeing the latest version.

Cerebral was intended to help accelerate the workflow of the model-experiment cycle, where experiment results need to be interpreted in the context of the current graph model, with the outcome that modifications to the model are proposed to refine it. In this case, experiments on cells produce measurements of gene expression levels for nodes in the graph using microarrays, but the particular technology used to create the quantitative data is not important for the purposes of visualization design.

Before Cerebral, the biologists were constrained to looking at a very limited region of the graph model, centered around a single biomolecule, because of the difficulty of creating comprehensible images through manual layout. Figure 4 shows an example of a local neighborhood around the `TLR4` biomolecule, with around 50 nodes. The goal of the Cerebral system was to allow them to interact with the entire immune system, which is an order of magnitude larger with over 700 nodes and 1200 edges, as shown in Figure 5.

3.2 Encoding: custom graph layout

Graph layout is a very heavily studied problem, with hundreds of papers over the past thirty years, and an ongoing annual Graph Drawing conference. Force-directed methods based on Fruchterman and Reingold[14] and hierarchical layouts based on Sugiyama[15] have become extremely popular, and a vast amount of work has gone into more sophisticated general graph layouts that scale up to larger datasets.[16]

Why, then, did we choose to create yet another layout algorithm? The difficulty is that the general graph layout problem is exactly and only to create a layout in the two-dimensional plane that best captures the topological structure of a set of nodes connected by links. In contrast, the needs of biologists engaged in the model-experiment cycle are to see biological knowledge, and the topological relationships within the systems biology graph model are only a subset of the bigger dataset. In this case, one of the most relevant aspects of additional knowledge was the location within the cell where the interaction between biomolecules occurs. Biological cells are divided into compartments by membranes, and interactions generally occur within a compartment. The location of the interaction is often known as part of the model, in the form of metadata for the node. Stylized hand-drawn diagrams, where subcellular location is spatially encoded along a vertical axis so that all of the items within a compartment are on the same large horizontal band, are very common in biology textbooks and papers. The key design decision was to create a biologically relevant layout method that is similar in spirit to these hand-drawn diagrams, where the vertical spatial position reveals the location inside the cell, but that runs completely automatically.

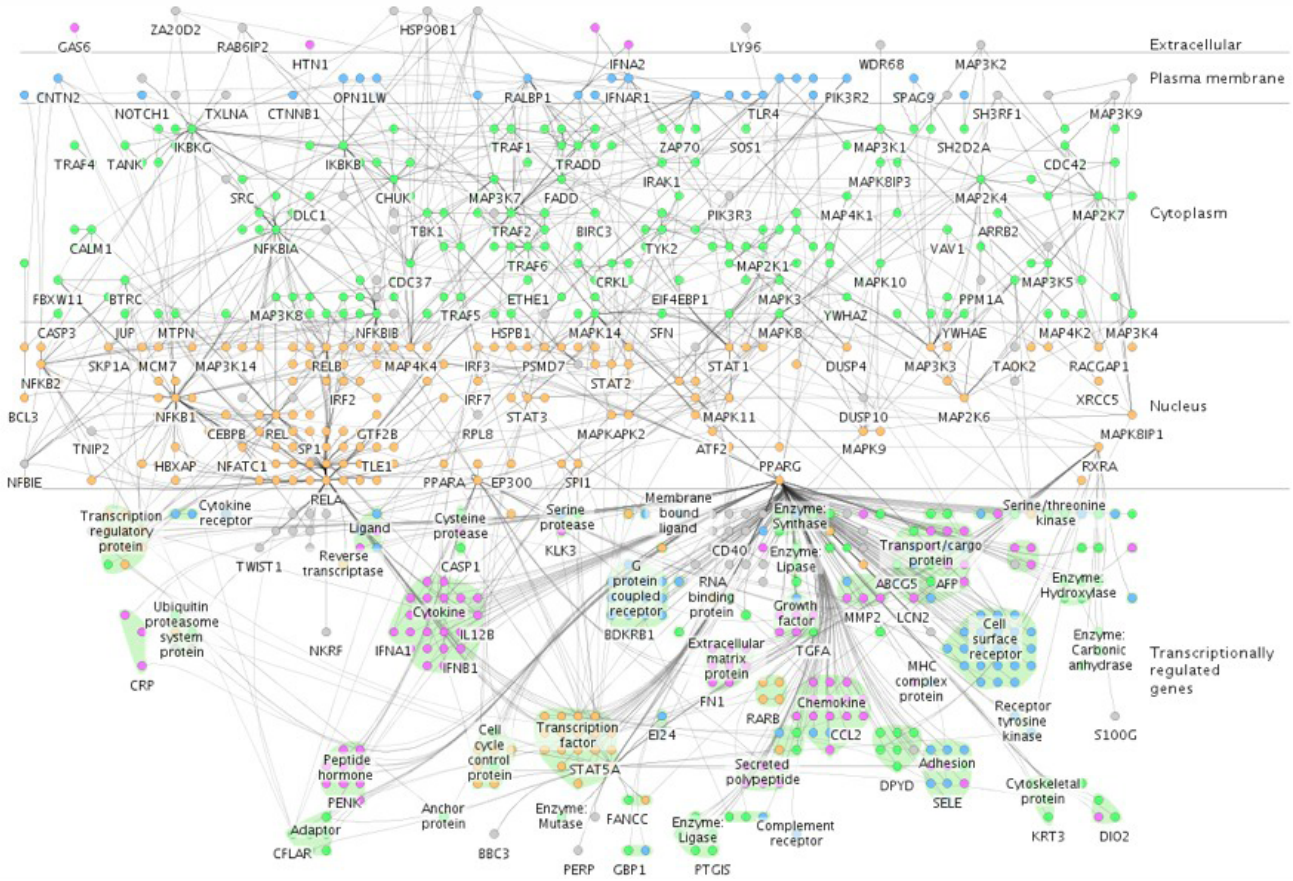

Figure 5. Entire immune system, with 760 nodes and 1263 edges.

3.3 Algorithm: simulated annealing

At the algorithmic level, we designed a new algorithm based on simulated annealing. Previous attempts along these lines[17–20] did not scale beyond small graphs of around 100 nodes because they had $O(V^3)$ algorithm their core, requiring time cubic in the number of vertices V. Our method[21] uses a fast discretization-based framework that achieves in $O(E\sqrt{V})$ time, where E is the number of edges.

3.4 Encoding/Interaction: small multiples

Cerebral uses an array of small multiple views, showing one graph instance for each experimental condition. These windows all share the same spatial layout, which is also used in the main graph interaction window. What differs between the windows is the coloring of the nodes, according to that condition. The main window either shows the coloring of the currently selected small multiple, or a view showing the difference between two of them using a color coding of their differences.

Another seemingly obvious possibility would be to use animation, especially since often the experimental data is a time series. However, this dataset is an excellent example of the problems with animation as a visual encoding mechanism for multi-frame dynamic data with complex changes at each step. Animation works well for spotting small changes between two complex scenes by flipping back and forth between just two frames, as in the blink comparator technique used by astronomers. It also works well for storytelling where the motion in each frame is highly choreographed so that the viewer's eye is drawn to exactly the right place at the right time.[22] However, in this case, global comparison is extremely difficult because so many things change all over the view at each step, and there are many steps. Creating a flipbook movie of the views in each small multiple illustrates the problem clearly. The viewer can focus attention on one node and notice the sequence of changing colors, but completely loses track of what happens anywhere else in the scene.

The limits of animation compared to side by side visual comparison have been documented in the research literature. The fundamental issue is that human memory has a very limited capacity, and so we do more poorly comparing the memory of what we saw before to what is visible at any moment, as opposed to quickly moving our eyes back and forth between simultaneously-visible multiple views. Tversky *et al.* present a very thorough meta-review of empirical studies on animation use and find that it does not facilitate in most cases; many seemingly positive results arose from studies that did not sufficiently control for the fact that animated views presented more information than static views.[23] Plumlee and Ware present a detailed example of the strengths of multiple-view interfaces versus zooming as dynamic change in a single view.[24] Robertson *et al.* present a careful empirical analysis of the use of Gapminder-style animation that comes to similar conclusions.[25]

Another alternative to small multiple views are glyphs, where information about the multiple conditions is embedded as a chart inside the visual representation for each node, as proposed by Westenberg *et al.*[26] While this approach works well when zoomed in to inspect a very small number of nodes, when the charts are clearly visible, it does not scale up to providing an overview for larger graphs. When zoomed out, the viewer cannot possibly see the details of within each chart, because there are so few pixels available for each node that only one value can be shown at each node, typically with color coding.

3.5 Encoding/Interaction: multiple views

In addition to small multiples, Cerebral also uses multiple views. The potentially confusing terminology for interfaces with more than one window is a historical artifact. In the language of Munzner's taxonomy,[27] the *multiple view* approach is where each view has a different visual encoding that shows different aspects of the dataset most clearly, and linked highlighting between is powerful because it shows when items that fall into a contiguous region in one view are also nearby or widely distributed in another view. (In contrast, with *small multiples* each view has the same visual encoding but for different datasets, typically with shared axes between frames so that comparison of spatial position between them is meaningful.)

The choice in Cerebral was to show two different visual encodings. In the graph view, spatial position is used to show topological structure of the systems biology graph, and measurement information is shown with color. Thus graph structure is primary, and measurement information is secondary. In the parallel coordinates view, measurement data is primary, since it is encoded with spatial position.

One seemingly obvious question is why the graph views need to be shown next to the measurements: why not simply show the measurements alone, as the new information gathered in the scientific experiment? The measurement data could support data-driven hypothesis generation, where clusters in gene expression would provide evidence for similar functions of those genes within the cell.

The answer is that the biologists consider purely data-driven clusters to be untrustworthy evidence with a strong chance of being an artifact because their data is intrinsically very noisy. Their experience is that the many clustering algorithms all give different results, so they do not trust any of them. Thus, the measured data needs to be evaluated in the context of the graph view: clusters must be further supported by corresponding to some information in the graph view with respect to previously known biological function, before they are worth investigating further. Thus, the design rationale in Cerebral was to provide linked highlighting between both a graph-primary and a measurement-primary view, simultaneously visible.

3.6 Validation

At the problem level, the Cerebral design was validated throughout the design cycle by working in collaboration with a specific target group of researchers, from the initial requirements gathering phase through iterative refinement of a series of interactive software prototypes.

Cerebral was released as an open-source plugin to the popular Cytoscape biological network visualization platform,[28] with an early version released in 2007 for the custom graph layout and a later version in 2008 that included support for multiple experimental conditions. Since several years have now elapsed, we can usefully consider the adoption rates as a downstream validation of our domain problem analysis.

Cerebral has indeed been adopted by the Hancock Lab innate immunity research group, our collaborators during its design process. It is featured prominently in their flagship InnateDB database,[29] and was cited in six of their other research papers.[30–35]

A literature search in January 2011 of papers that cite Cerebral yielded 15 biology or bioinformatics research papers from other research groups with whom we have no direct connection, where Cerebral was used as part of the biology research process.[36–50] This count includes papers that mention Cerebral in their methods section or have research-content figures generated with it. In contrast, we classified 18 other citations as being from algorithm-oriented or survey-style papers, and do not count them as adoption evidence.

At the visual encoding and interaction levels, one validation method in the original paper [12] was the qualitative discussion of system usage and visual results in the case studies section. Another validation method was the careful justification of the design rationale, both in that publication and in further detail in this paper. At the algorithm level, Cerebral was validated through both complexity analysis and benchmark timings.[12,21,51]

4. PATHLINE

The Pathline system [13] was joint work with Miriah Meyer, Bang Wong, Mark Styzynski, and Hanspeter Pfister. It was designed in collaboration with researchers studying metabolism in yeast at the Regev lab of the Broad Institute. Figure 6 shows the Pathline interface, which shows information on metabolic pathways in the linearized pathway view on the left, about gene expression in the curvemap view on the right, and phylogeny information with the tree in the middle.

We will discuss design choices at three of the levels, starting the problem of supporting comparative functional genomics and continuing with the abstractions of data and tasks. We will discuss two visual encoding level choices in detail, the linearized pathway and the curvemap views. We will conclude the Pathline example with a discussion of validation.

4.1 Problem: comparative functional genomics

The problem of functional genomics is to understand how genes work together to perform different functions within a cell. Biologists measure gene expression levels indicating how much a gene is turned on or off. They make these measurements for many genes and for many samples, where the samples could be time points, tissue types, or species. Biologists also have a graph model, similar to the previous example, but focused on cell function in the metabolic network. In this graph, the nodes are metabolites, and links represent genes whose products catalyze chemical reactions. The biologists do not typically study the entire metabolic network at once, but filter it by breaking it down into known pathways of around a dozen reactions each. They then study the gene and metabolite levels for only a handful of these pathways simultaneously.

In comparative functional genomics, the further question is how to understand how gene interactions vary across different species. The biologists thus must compare all of this data across multiple species. The ancestral relationships between species are important for their analysis, so they must consider the phylogenetic tree of evolutionary relationships between them rather than just considering species as an unordered list.

The yeast researchers thus needed to see at multiple genes at multiple time points, across multiple pathways, all across multiple related species. Previous tools could only show them a subset of this complex heterogeneous dataset at once.

4.2 Abstraction: data and tasks

We first discuss the data abstractions. A metabolic pathway was considered to be a small directed graph of around one dozen nodes, representing metabolites, that could contain cycles or branches. The total number of pathways of interest was between 10 and 50, but only a small number needed to be seen at once. The gene and metabolite levels data was treated as an abstract 3D table with a single quantitative value at each cell. One table axis was the 6000 genes and 140 metabolites, another was the 6 time points at which biologically interesting events occurred, and the third was the 14 species of yeast under study. The phylogeny of evolutionary relationships was treated as a binary tree. Finally, the biologists computed a similarity score for each single gene

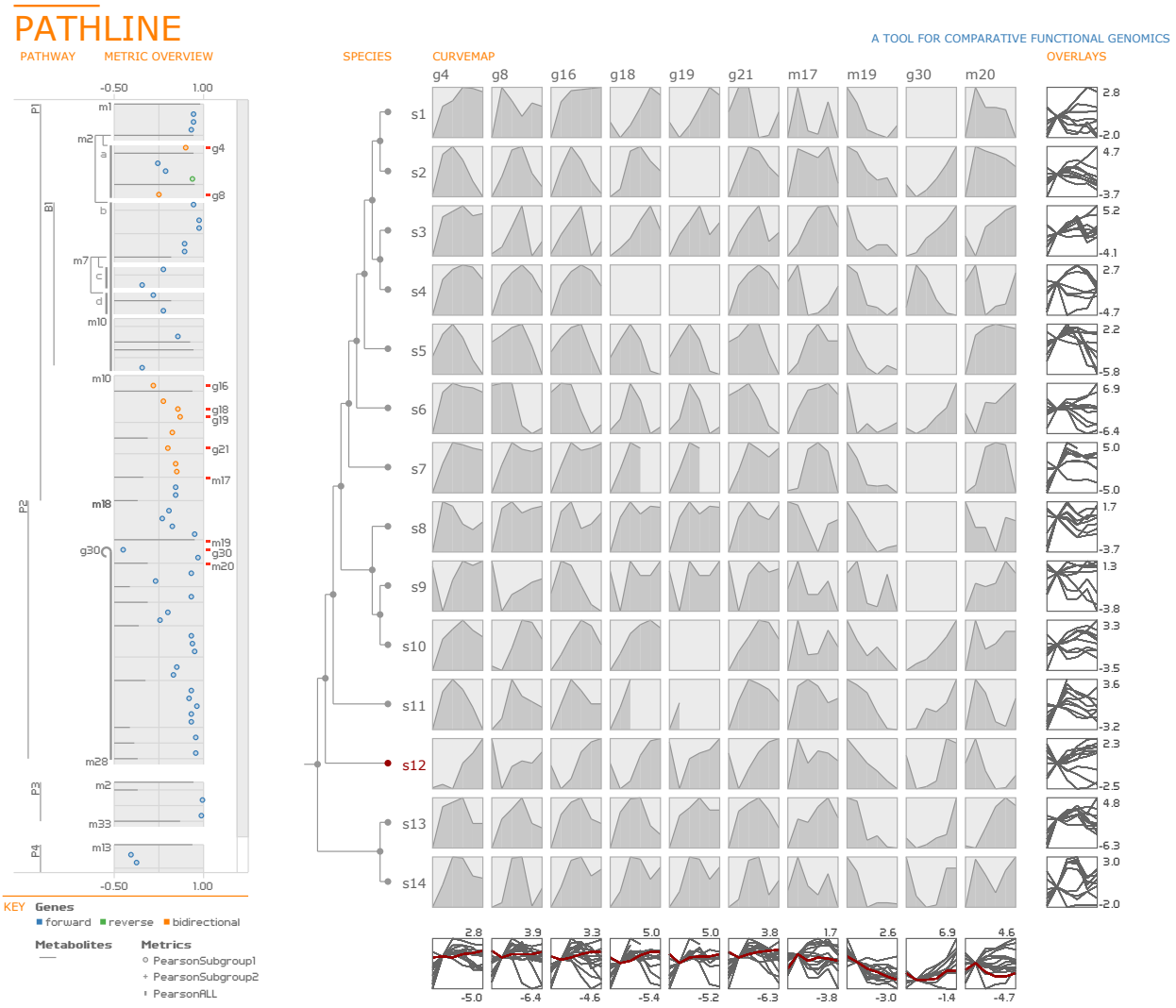

Figure 6. The Pathline interface features a linearized pathway view on the left, a curvemap view on the right, and a phylogenetic tree in between.

or metabolite across multiple species. Their dataset included multiple similarity scores for each, computed with different algorithms.

The task abstractions that we identified after extensive discussions with the biologists were to study expression data as a time series, to compare a limited number of time series, to compare similarity scores along a small set of pathways, and the comparison of multiple similarity scores.

4.3 Encoding: linearized pathway

The common practice with node-link graphs is to use two-dimensional planar position for a graph layout that emphasizes topological structure. Even in the Cerebral example above, where one of those dimensions was partially reserved for showing subcellular position, the emphasis was on showing topological structure.

However, the topological structure of these fairly short pathways is not complex: they are most often linear chains, with occasional cycles and branches. Moreover, the set of pathways under study is small enough that many of them are quite familiar to the researchers. We determined after task analysis that while the topological

structure of the pathways was of interest, the quantitative measurements at each pathway node were even more important. Mackinlay's principle of effectiveness dictates that the most highly ranked visual cue should be used for the most important dimension of the data.[52] A key principle of visual encoding is that spatial position is the most highly ranked visual cue. In particular, spatial position within a common aligned frame is the most accurately perceived visual cue for quantitative data.[53] We thus use spatial position within a shared frame to show the quantitative similarity scores as the primary aspect of the linearized pathway view, and relegate topological structure to secondary information, shown with with stylized marks that indicate unrolled branches and cycles.

The horizontal axis shows the similarity score values, and the vertical axis shows the position of the item within the pathway. Multiple pathways are abutted vertically, retaining the common frame. Gene levels are encoded with circles while metabolites are shown with bars, creating visual layers that can be selectively attended to or seen together. The different similarity scores can be encoded with shapes along the same common scale, allowing high-precision visual comparison.

4.4 Encoding: curvemap

The extremely common practice in biological applications is to show gene levels using heatmaps,[54] where color is used to visually encode the quantitative information. However, as in the discussion above, people can make perceptual judgements about curve shape more accurately than those about color changes.[55] Moreover, our task analysis revealed that the biologists discussed their tasks using shape-oriented language like *peaks* and *valleys*.

We thus proposed the curvemap view, with a matrix of filled, framed line charts rather than colored blocks, in order to enhance shape perception. In this case the rows are species, following the leaves of the phylogenetic tree, and columns are genes or metabolites. Overlays to the side and bottom allow direct comparison between curve shapes, to enhance trend perception. A column in the curvemap view provides the details underlying a circle or bar in the linearized pathway view; clicking in that view controls which columns are shown in the curvemap.

4.5 Validation

As above, the problem-level validation of Pathline was carried out by working in close collaboration with a target group through the entire design cycle. An adoption success is that it remains in daily use by the yeast researchers. The tool was released as open source only six months ago, so it is not surprising that there are not yet any citations of use from the biological community. At the abstraction level, we have anecdotal evidence that the tool allowed the researchers to verify previous analyses far more quickly, and to make new discoveries. Our collaborators are actively working on papers that resulted from discoveries made using this tool. Again, as above, the validation at the encoding and interaction levels included both qualitative discussion of case studies and justification of the design rationale.[13] No explicit validation was carried out at the algorithmic level beyond the baseline of providing a working software system, since no claims of contribution were made at that level.

5. CONCLUSION

We have elaborated on the principles of information visualization presented in the nested model framework[3] by providing a detailed walkthrough of decision decisions for two biological applications, Cerebral[12] and Pathline.[13] Both feature biological networks, but as part of a larger context of relevant information rather than as the only aspect of interest. We discussed the reasons for making design decisions that diverged from common approaches in previous work, through a combination of the careful abstraction of tasks and data, and a knowledge of visual encoding principles.

REFERENCES

[1] Zhang, J., "The nature of external representations in problem solving," *Cognitive Science* **21**(2), 179–217 (1997).

[2] Anscombe, F., "Graphs in statistical analysis," *American Statistician* **27**, 17–21 (Feb. 1973).

[3] Munzner, T., "A nested model for visualization design and validation," *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2009)* **15**(6), 921–928 (2009).

[4] Munzner, T., Hoffman, E., Claffy, K., and Fenner, B., "Visualizing the global topology of the MBone," in [*Proceedings of the 1996 IEEE Symposium on Information Visualization*], 85–92 (1996).

[5] Munzner, T., Guimbretière, F., and Robertson, G., "Constellation: A visualization tool for linguistic queries from mindnet," in [*Proceedings of the 1999 IEEE Symposium on Information Visualization*], 132–135 (1999).

[6] Lam, H., Russell, D., Tang, D., and Munzner, T., "Session Viewer: Visual Exploratory Analysis of Web Session Logs," in [*Proc. IEEE Symposium on Visual Analytics Science and Technology (VAST'07)*], 147–154 (2007).

[7] Munzner, T., "Exploring large graphs in hyperbolic space," *IEEE Computer Graphics and its Applications* **18**, 18–23 (July/August 1998).

[8] McLachlan, P., Munzner, T., Koutsofios, E., and North, S., "LiveRAC: interactive visual exploration of system management time-series data," in [*Proc. ACM Conf. on Human Factors in Computing Systems (CHI)*], 1483–1492 (2008).

[9] Meyer, M., Munzner, T., and Pfister, H., "MizBee: A multiscale synteny browser," *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2009)* **15**(6), 897–904 (2009).

[10] Meyer, M., Munzner, T., dePace, A., and Pfister, H., "MulteeSum: A tool for exploring space-time expression data," *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2010)* **16**(6), 908–917 (2010).

[11] Munzner, T., Guimbretière, F., Tasiran, S., Zhang, L., and Zhou, Y., "TreeJuxtaposer: Scalable tree comparison using Focus+Context with guaranteed visibility," *ACM Transactions on Graphics (Proc. SIGGRAPH)* **22**(3), 453–462 (2003).

[12] Barsky, A., Munzner, T., Gardy, J., and Kincaid, R., "Cerebral: Visualizing multiple experimental conditions on a graph with biological context," *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2008)* **14**(6), 1253–1260 (2008).

[13] Meyer, M., Wong, B., Styczynski, M., Munzner, T., and Pfister, H., "Pathline: A tool for comparative functional genomics," *Computer Graphics Forum (Proc. EuroVis 10)* **29**(3), 1043–1052 (2010).

[14] Fruchterman, T. M. J. and Reingold, E. M., "Graph drawing by force-directed placement.," *Software - Practice and Experience* **21**, 1129–1164 (November 1991).

[15] Sugiyama, K. and Misue, K., "Visualization of structural information: Automatic drawing of compound digraphs," *IEEE Trans. on Systems, Man, and Cybernetics* **21**(4), 876–892 (1991).

[16] Archambault, D., Munzner, T., and Auber, D., "TopoLayout: Multilevel graph layout by topological features," *IEEE Trans. on Visualization and Computer Graphics* **13**, 305–317 (March/April 2007).

[17] Davidson, R. and Harel, D., "Drawing graphs nicely using simulated annealing," *ACM Trans. on Graphics* **15**(4), 301–331 (1996).

[18] Kato, M., Nagasaki, M., Doi, A., and Miyano, S., "Automatic drawing of biological networks using cross cost and subcomponent data," *Genome Inform.* **16**(2), 22–31 (2005).

[19] Kojima, K., Nagasaki, M., Jeong, E., Kato, M., and Miyano, S., "An efficient grid layout algorithm for biological networks utilizing various biological attributes," *BMC Bioinformatics* **8**, 76+ (March 2007).

[20] Li, W. and Kurata, H., "A grid layout algorithm for automatic drawing of biochemical networks," *Bioinformatics* **21**, 2036–2042 (May 2005).

[21] Barsky, A., Gardy, J. L., Hancock, R. E., and Munzner, T., "Cerebral: a Cytoscape plugin for layout of and interaction with biological networks using subcellular localization annotation," *Bioinformatics* **23**(8), 1040–1042 (2007).

[22] Lasseter, J., "Principles of traditional animation applied to 3D computer animation," *Computer Graphics (Proc. SIGGRAPH 87)* **21**(4), 35–44 (1987).

[23] Tversky, B., Morrison, J., and Betrancourt, M., "Animation: Can it facilitate?," *International Journal of Human Computer Studies* **57**(4), 247–262 (2002).

[24] Plumlee, M. and Ware, C., "Zooming versus multiple window interfaces: Cognitive costs of visual comparisons," *Proc. ACM Trans. on Computer-Human Interaction (ToCHI)* **13**(2), 179–209 (2006).

[25] Robertson, G., Fernandez, R., Fisher, D., Lee, B., and Stasko, J., "Effectiveness of animation in trend visualization," *IEEE Trans. on Visualization and Computer Graphics (Proc. InfoVis08)* **14**(6), 1325–1332 (2008).

[26] Westenberg, M. A. et al., "Interactive visualization of gene regulatory networks with associated gene expression time series data," in [*Visualization in Medicine and Life Sciences, Visualization and Mathematics*], 293–312 (2008).

[27] Munzner, T., "Visualization (Chapter 27)," in [*Fundamentals of Graphics*], 675–707, AK Peters, 3rd ed. (2009).

[28] Shannon, P., Markiel, A., Ozier, O., Baliga, N., Wang, J., Ramage, D., Amin, N., Schwikowski, B., and Ideker, T., "Cytoscape: A software environment for integrated models of biomolecular interaction networks," *Genome Research* **13**, 2498–2504 (2003).

[29] Lynn, D. J., Winsor, G. L., Chan, C., Richard, N., Laird, M. R., Barsky, A., Gardy, J. L., Roche, F. M., Chan, T. H. W., Shah, N., Lo, R., Naseer, M., Que, J., Yau, M., Acab, M., Tulpan, D., Whiteside, M. D., Chikatamarla, A., Mah, B., Munzner, T., Hokamp, K., Hancock, R. E. W., and Brinkman, F. S. L., "InnateDB: facilitating systems-level analyses of the mammalian innate immune response," *Molecular Systems Biology* **4**(218) (2008).

[30] Brown, K. L., Cosseau, C., Gardy, J. L., and Hancock, R. E., "Complexities of targeting innate immunity to treat infection," *Trends in Immunology* **28**(6), 260–266 (2007).

[31] Gardy, J. L., Lynn, D. J., Brinkman, F. S., and Hancock, R. E., "Enabling a systems biology approach to immunology: focus on innate immunity," *Trends in Immunology* **30**(6), 249 – 262 (2009).

[32] Lynn, D. J., Gardy, J. L., Hancock, R. E., and Brinkman, F. S., "Systems-level analyses of the mammalian innate immune response," in [*Systems Biology for Signaling Networks*], 531–560, Springer (2010).

[33] Lynn, D. J., Chan, C., Naseer, M., Yau, M., Lo, R., Sribnaia, A., Ring, G., Que, J., Wee, K., Winsor, G. L., Laird, M. R., Breuer, K., Foroushani, A. K., Brinkman, F. S., and Hancock, R. E., "Curating the innate immunity interactome," *BMC Systems Biology* **4** (Aug. 2010).

[34] Mayer, M. L., Sheridan, J. A., Blohmke, C. J., Turvey, S. E., and Hancock, R. E. W., "The *pseudomonas aeruginosa* autoinducer 3o-c12 homoserine lactone provokes hyperinflammatory responses from cystic fibrosis airway epithelial cells," *PLoS ONE* **6**, e16246 (01 2011).

[35] Mookherjee, N., Rehaume, L. M., and Hancock, R. E., "Cathelicidins and functional analogues as antisepsis molecules," *Informa Healthcare* **11**(8), 993–1004 (2007).

[36] Diez, D., Wheelock, A. M., Goto, S., Haeggström, J. Z., Paulsson-Berne, G., Hansson, G. K., Hedin, U., Gabrielsen, A., and Wheelock, C. E., "The use of network analyses for elucidating mechanisms in cardiovascular disease," *Molecular BioSystems* **6**(2), 289–304 (2010).

[37] Dyer, M. D., Murali, T. M., and Sobral, B. W., "The landscape of human proteins interacting with viruses and other pathogens," *PLoS Pathogend* **4**, e32 (02 2008).

[38] Dyer, M. D., *Pathosystems Biology: Computational Prediction and Analysis of Host-Pathogen Protein Interaction Networks*, PhD thesis, Virginia Tech Department of Genetics, Bioinformatics, and Computational Biology (2008).

[39] Echeverría, P. C., Mazaira, G., Erlejman, A., Gomez-Sanchez, C., Pilipuk, G. P., and Galigniana, M. D., "Nuclear import of the glucocorticoid receptor-hsp90 complex through the nuclear pore complex is mediated by its interaction with nup62 and importin β," *Molecular and Cellular Biology* **29**(17), 4788–4797 (2009).

[40] Echeverría, P. C., Figueras, M. J., Vogler, M., Kriehuber, T., de Miguel, N., Deng, B., Dalmasso, M. C., Matthews, D. E., Matrajt, M., Haslbeck, M., Buchner, J., and Angel, S. O., "The hsp90 co-chaperone p23 of toxoplasma gondii: Identification, functional analysis and dynamic interactome determination," *Molecular and Biochemical Parasitology* **172**(2), 129–140 (2010).

[41] He, L., Sun, Y., Takemoto, M., Norlin, J., Tryggvason, K., Samuelsson, T., and Betsholtz, C., "The glomerular transcriptome and a predicted protein-protein interaction network," *J. Am. Soc. Nephrol.* **19**(2), 260–268 (2008).

[42] Li, X., Cai, H., Xu, J., Ying, S., and Zhang, Y., "A mouse protein interactome through combined literature mining with multiple sources of interaction evidence," *Amino Acids* **38**(4), 1237–1252 (2010).

[43] Mitsuhashi, T., Warita, K., Tabuchi, Y., Takasaki, I., Kondo, T., Sugawara, T., Hayashi, F., Wang, Z.-Y., Matsumoto, Y., Miki, T., Takeuchi, Y., Ebina, Y., Yamada, H., Sakuragi, N., Yokoyama, T., Nanmori, T., Kitagawa, H., Kant, J. A., and Hoshi, N., "Global gene profiling and comprehensive bioinformatics analysis of a 46,xy female with pericentric inversion of the y chromosome," *Congenital Anomalies* **50**, 40–51 (2010).

[44] Muller, F.-J., Laurent, L. C., Kostka, D., Ulitsky, I., Williams, R., Lu, C., Park, I.-H., Rao, M. S., Shamir, R., Schwartz, P. H., Schmidt, N. O., and Loring, J. F., "Regulatory networks define phenotypic classes of human stem cell lines," *Nature* **455**, 401–405 (2008).

[45] Newman, A. M. and Cooper, J. B., "AutoSOME: a clustering method for identifying gene expression modules without prior knowledge of cluster number," *BMC Bioinformatics* **11**, 117 (2010).

[46] Padgett, K. A., Lan, R. Y., Leung, P. C., Lleo, A., Dawson, K., Pfeiff, J., Mao, T. K., Coppel, R. L., Ansari, A. A., and Gershwin, M. E., "Primary biliary cirrhosis is associated with altered hepatic microrna expression," *Journal of Autoimmunity* **32**(3-4), 246 – 253 (2009).

[47] Squires, R. B., *An Influenza Virus Molecular Infection Model And Discrete Event, Stochastic Simulation*, Master's thesis, University of Texas at Arlington Department of Computer Science and Engineering (2008).

[48] Tari, L., Anwar, S., Liang, S., Hakenberg, J., and Baral, C., "Synthesis of pharmacokinetic pathways through knowledge acquisition and automated reasoning," in [*Proceedings of the Pacific Symposium on Biocomputing*], 465–476 (2010).

[49] Wang, Y.-C. and Chen, B.-S., "Integrated cellular network of transcription regulations and protein-protein interactions," *BMC Systems Biology* **4**, 20 (2010).

[50] Zhu, W., Yang, L., and Du, Z., "Layered functional network analysis of gene expression in human heart failure," *PLoS ONE* **4**, e6288 (07 2009).

[51] Barsky, A., *Cerebral: Visualizing Multiple Experimental Conditions on a Graph with Biological Context*, Master's thesis, University of British Columbia, Dept. Computer Science, Vancouver (June 2008).

[52] Mackinlay, J., "Automating the design of graphical presentations of relational information," *ACM Trans. on Graphics (TOG)* **5**(2), 110–141 (1986).

[53] Cleveland, W. S. and McGill, R., "Graphical perception: Theory, experimentation, and application to the development of graphical methods," *Journal of the American Statistical Association* **79**(387), 531–554 (1984).

[54] Wilkinson, L. and Friendly, M., "The history of the cluster heat map," *The American Statistician* **63**(2), 179–184 (2009).

[55] Lam, H., Munzner, T., and Kincaid, R., "Overview use in multiple visual information resolution interfaces," *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 2007)* **13**(6), 1278–1285 (2007).