

The FlowVizMenu and Parallel Scatterplot Matrix: Hybrid Multidimensional Visualizations for Network Exploration

Christophe Viau, Michael J. McGuffin, Yves Chiricota, and Igor Jurisica

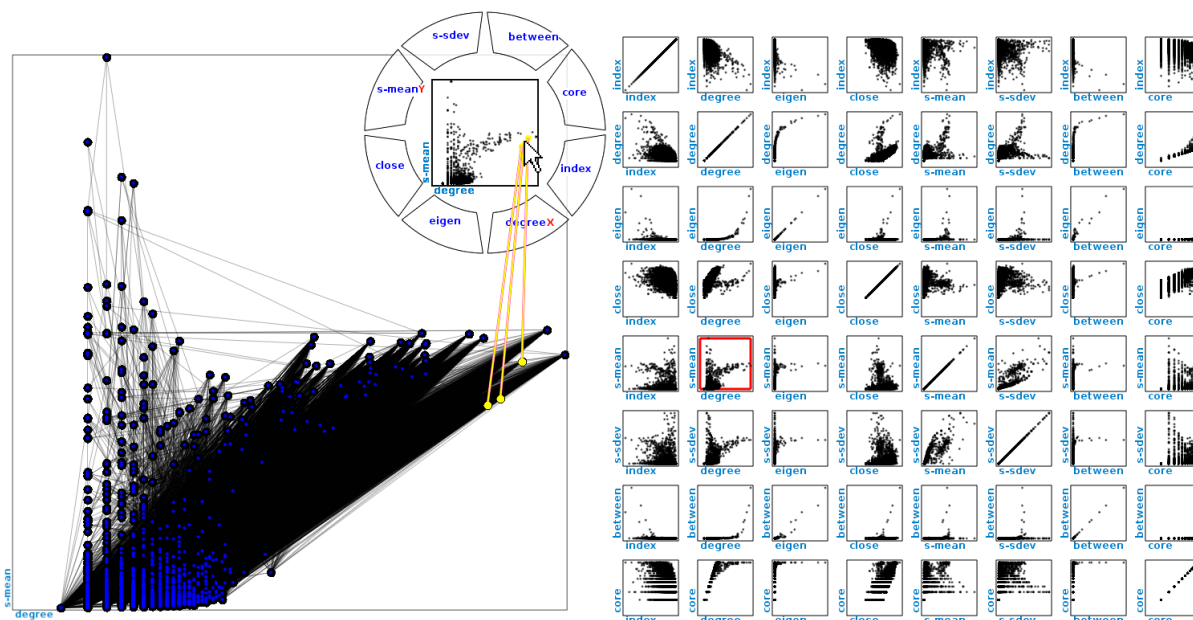


Fig. 1. The user interface. At left: the node-link diagram, here with nodes positioned according to an attribute-driven layout, i.e., adopting their corresponding positions within a $\text{degree} \times \text{s-mean}$ scatterplot. Top middle: the FlowVizMenu is popped up and contains the same scatterplot. Fluid gestures within the menu select dimensions to drive the attribute-driven layout with smoothly animated transitions. At right: the P-SPLOM, here showing a SPLOM of the nodes' metrics.

Abstract—A standard approach for visualizing multivariate networks is to use one or more multidimensional views (for example, scatterplots) for selecting nodes by various metrics, possibly coordinated with a node-link view of the network. In this paper, we present three novel approaches for achieving a tighter integration of these views through hybrid techniques for multidimensional visualization, graph selection and layout. First, we present the FlowVizMenu, a radial menu containing a scatterplot that can be popped up transiently and manipulated with rapid, fluid gestures to select and modify the axes of its scatterplot. Second, the FlowVizMenu can be used to steer an attribute-driven layout of the network, causing certain nodes of a node-link diagram to move toward their corresponding positions in a scatterplot while others can be positioned manually or by force-directed layout. Third, we describe a novel hybrid approach that combines a scatterplot matrix (SPLOM) and parallel coordinates called the Parallel Scatterplot Matrix (P-SPLOM), which can be used to visualize and select features within the network. We also describe a novel arrangement of scatterplots called the Scatterplot Staircase (SPLOS) that requires less space than a traditional scatterplot matrix. Initial user feedback is reported.

Index Terms—Interactive graph drawing, network layout, attribute-driven layout, parallel coordinates, scatterplot matrix, radial menu.

1 INTRODUCTION

- *Christophe Viau is with École de technologie supérieure, Montreal, Canada, E-mail: christopheviau@gmail.com.*
- *Michael J. McGuffin is with École de technologie supérieure, Montreal, Canada, E-mail: michael.mcguiffin@etsmtl.ca.*
- *Yves Chiricota is with Université du Québec à Chicoutimi, Chicoutimi, Canada, E-mail: ychirico@uqac.ca.*
- *Igor Jurisica is with Ontario Cancer Institute, PMH/UHN, Toronto, Canada, E-mail: juris@ai.utoronto.ca.*

Manuscript received 31 March 2010; accepted 1 August 2010; posted online 24 October 2010; mailed on 16 October 2010.

For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

The most common approach for visualizing a network is a node-link diagram, for which there are many layout algorithms [11]. Unfortunately, some networks are so complicated that it may be impossible to give them a layout that makes the most important nodes, their connections, and their immediate neighbors clearly visible. For example, in our own work with biological networks, we often have individual nodes with over 300 neighbors; just laying out these neighbors in a clear way is challenging. In addition, a single, automatically generated layout may not be appropriate for all situations, leading to users often manually adjusting and repositioning certain nodes in the node-link diagram. When the nodes of interest to the user are not immediately visible, some indirect means of searching and selecting them is required. For example, a spreadsheet interface (such as that in [5, 4]) might list all nodes and allow them to be sorted by name, degree, clustering coefficient, etc. and selected. Another possibility is to have a scatterplot, of clustering coefficient versus degree for example, within

which the user may select nodes [9].

Taking this idea further, for any given network, we could compute various metrics associated with each node (such as degree, clustering coefficient, betweenness centrality, etc.) and we may also have various attributes associated with each node (protein name, biological function, cellular localization, etc.) Such multidimensional data motivates the use of standard multidimensional visualization techniques, in particular: scatterplot matrices (SPLOMs) [18] and parallel coordinate plots (PCPs) [12, 20, 35]. Recent research by Bezerianos et al. [4] takes such an approach, using a SPLOM to visualize a multivariate graph. In their system, the SPLOM serves as a kind of overview of the data, and is coordinated with a single zoomed-in scatterplot that serves as the focus. Links between nodes are drawn on top of the scatterplot to reveal the graph’s structure. A ScatterDice-style interface [13] allows for transitions of dimensions through 3D rotation.

Our work further explores the design space of network visualizations that incorporate multidimensional visualization techniques. We have developed a novel combination of a node-link diagram, a SPLOM, and PCPs (Figures 1 and 2). The layout of our node-link diagram can be modified manually, or through force-directed layout, or attribute-driven layout, or a mixture of these. Our SPLOM and PCPs are combined into a Parallel Scatterplot Matrix (P-SPLOM) that affords fluid transition between scatterplots, 3D PCPs, and normal (2D) PCPs using 3D rotation. We also present a novel popup widget that enables rapid, fluid gestures to select within scatterplots and modify the layout of the node-link diagram. Together, these techniques provide a greater variety, and a tighter integration, of visualizations of the network than has been previously possible.

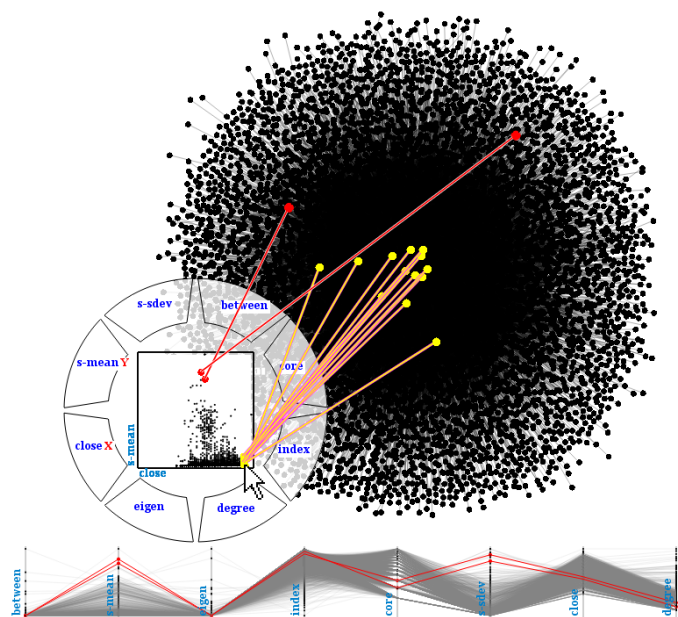


Fig. 2. Selecting (in red) and brushing (in yellow) within the FlowVizMenu’s scatterplot shows linking with the node-link diagram. Along the bottom of this screenshot, the P-SPLOM has been converted to a single row PCP, and selected nodes are shown as red polylines.

Our contributions are (1) a novel popup widget for manipulating multidimensional visualizations called the FlowVizMenu; (2) a technique for mixing attribute-driven layout with force-directed and manual layout of nodes within a node-link diagram; (3) a novel integration of SPLOMs, PCPs, and 3D PCPs called the P-SPLOM; (4) an investigation of the tradeoffs of different orderings of the axes within a P-SPLOM; and (5) a novel arrangement of scatterplots called the Scatterplot Staircase (SPLOS). We also report initial user feedback.

2 RELATED WORK

Recent work on multivariate graph visualization includes [34, 2, 4]. Of these, the most similar to our current work is GraphDice [4], in which there are two main views: the SPLOM, and a single zoomed-in scatterplot. The nodes in the scatterplot are always positioned according to the selected dimensions, such as degree, centrality, etc. — this could be called attribute-driven layout. Two special dimensions are the x and y positions of the nodes in a force-directed layout. Selecting these x and y positions as the dimensions for the scatterplot results in a “scatterplot” showing the force-directed layout of the graph. Our work differs from GraphDice in several respects: we allow mixing of manual, force-directed, and attribute-driven layout; we allow the SPLOM to be rearranged in many different ways, as discussed in Section 6; our SPLOM can also be rotated in 3D to yield 3D parallel coordinate plots (PCPs) or normal (2D) PCPs. Our FlowVizMenu also differs from the ControlMenu used in [4] to rotate dimensions, as discussed in Section 4.

Our system uses parallel coordinate plots (PCPs) [20, 35]. 3D variants of PCPs have been proposed before, including [15, 21], and, most relevant to our work, [14, 28]. The “Cube” of [14], and 3D parallel coordinates of [28], are both essentially sequences of parallel planes, with each plane containing a scatterplot. This can be equated with taking normal 2D PCPs and replacing each axis with a scatterplot on a plane. The links connecting corresponding points on consecutive planes could be compared to the links between planes in VisLink [10]. We will refer to this 3D variant of PCPs as simply 3D PCPs. The P-SPLOM we present in this work is a novel unification of SPLOMs, PCPs, and 3D PCPs, achieved through a simple 3D rotation of the scatterplots.

Hybrid multidimensional visualizations have also been proposed before, including combinations of PCPs and scatterplots [27, 39, 30, 19] (see also [38] for a similar idea) and parallel coordinates displayed within a matrix [1]. Our P-SPLOM contributes one additional hybrid multidimensional visualization, with tradeoffs compared to previous work. For example, the scatterplot-PCP hybrid of [19] displays both the points of scatterplots and the polylines of a PCP at the same time, but at an increased cost in screen space. On the other hand, our P-SPLOM allows users to only view scatterplots, or only view a PCP, or view a 3D PCP showing both the points within each (partially rotated) scatterplot and the polylines of the PCP.

3 METRICS USED

In principle, any number of attributes (protein name, biological function, cellular localization, etc.) associated with the nodes could be used as dimensions in our system. Our system does not require the network to have any attributes, however, as several metrics are defined and computed for each node from the network structure itself. These include `index`: a unique integer identifying the node (although this dimension often has little meaning, its use within a scatterplot can help de-occlude points to reveal a distribution); `degree`: the number of neighbors of the node; `between`, `close`, `eigen`: the betweenness, closeness, and eigenvector centralities, respectively¹; `cluster`: the clustering coefficient; `core`: the shell associated with the k -core decomposition of the network [37]; `s-mean` and `s-sdev`: the average and standard deviation, respectively, of the *strength* (as defined in [3]) of the edges adjacent on the node.

4 THE FLOWVIZMENU

The human-computer interaction (HCI) community has proposed several popup widgets that afford a gestural style of interaction [8, 22, 26, 17, 23], some of which have been applied to interactive visualization [25]. Popup widgets have several advantages: they require no screen space when not in use; they eliminate the need to travel back and forth between a work area and menu bars or panels of widgets located in the periphery; and they can also be invoked by holding down a button, keyboard key, or stylus tip, resulting in kinesthetic feedback that

¹<http://en.wikipedia.org/wiki/Centrality>

helps avoid mode errors [29] and helps to integrate the selection of arguments into a single phrase or gesture [7].

The FlowVizMenu² is a novel popup widget that contains a multidimensional visualization, and allows dimensions to be selected through outward and inward motions. Figure 3 shows the variant we implemented, which displays a single scatterplot at a time. The menu is popped up with a keyboard key, after which the pointing device may brush over the scatterplot (Figure 2). The user may also select dimensions for the scatterplot by stroking outward or inward, for x and y respectively. This allows the user to quickly switch between, and compare, scatterplots. The outward-inward motions were inspired by those in the original FlowMenu [17]. These quick, continuous gestures have been found to yield good selection performance [16]. When the pointer crosses over the name of a dimension, a smooth transition occurs, showing the new dimension rotate into the old dimension’s place. The FlowVizMenu’s rotation transition is similar to that in [13, 4] except that we use an orthographic projection instead of a perspective projection, to maintain the position of the points along the unchanging axis. Furthermore, when the pointer crosses over the name of a dimension, the user may “scrub” to control the progression of the transition. This scrubbing functionality is comparable to the way a Control Menu [26] works, in that it uses the mouse drag to control a continuous parameter. Thus, the FlowVizMenu can be seen as a hybrid between a FlowMenu and a Control Menu that furthermore incorporates a visualization. Teoh et al. [31] also proposed a radial layout of small visualizations to be used for navigation, however their radial layout is not within a popup widget.

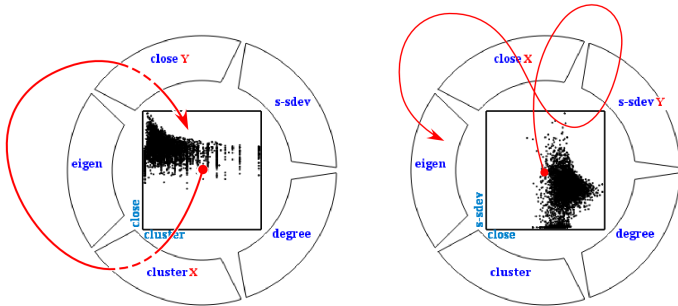


Fig. 3. Our implemented FlowVizMenu. *Left*: the $cluster \times close$ scatterplot was selected by stroking outward through the dimension for the horizontal axis and then inward through the dimension for the vertical axis. As the user strokes over the dimension names (dashed parts of the red arrow), the scatterplot rotates away from the previously selected axis and towards the new one. The user may “scrub” over the dimension name to slow down or replay this rotation. *Right*: Drawing a figure-8 gesture allows the user to rotate between two different scatterplots, in this case toggling between $s-sdev$ and $eigen$ on the vertical axis while maintaining $close$ on the horizontal axis. This allows for comparison of the two scatterplots within a relatively small space.

We can also compare our implemented FlowVizMenu with the control menu used in GraphDice [4]. Both are used to transition between dimensions in a scatterplot, and both allow for scrubbing. However, because GraphDice’s control menu only makes use of outward motions, it contains 2 copies of every dimension: one for the scatterplot’s horizontal axis, and another for the vertical axis. Furthermore, the menu items in GraphDice’s control menu only cover half a circle. The end result is that the dimensions in our FlowVizMenu each cover an angle that is four times larger, enabling easier and faster selection. Our FlowVizMenu also differs in that it *contains* a visualization, whereas GraphDice’s control menu is used to control an underlying visualization. As shown in Figure 2, this means the user can perform brushing and linking for coordination with other views without leaving the

²We initially considered calling this widget the MatchWheel, because MatchWheel + HotBox form a chiasm with MatchBox + HotWheel, two well known toy car brands.

FlowVizMenu.

In our FlowVizMenu, each repeated outward-inward motion will normally replace the two dimensions in the scatterplot with new dimensions. Hence, a repeated figure-8 motion (Figure 3, right) cycles between two scatterplots. However, it also occurred to us that repeated outward-inward motion could be useful for *accumulating* dimensions. In our implementation, holding down the shift key during motions causes the dimensions to be accumulated along the axes using principle component analysis (PCA). For example, holding down Shift, and moving out through dimension A, in through B, out through C, and in through D, will cause the system to compute a PCA projection from $A \times C$ to the horizontal axis, and a separate PCA projection from $B \times D$ to the vertical axis.

Another way that dimensions could be “accumulated” during repeated motions is in building up a PCP. We designed (but did not implement) this idea, as shown in Figure 4, where the accumulated dimensions result in a 1D histogram, then a 2D scatterplot, then a 3- (or more) axis PCP. Yet another design (also not implemented) is shown in Figure 5.

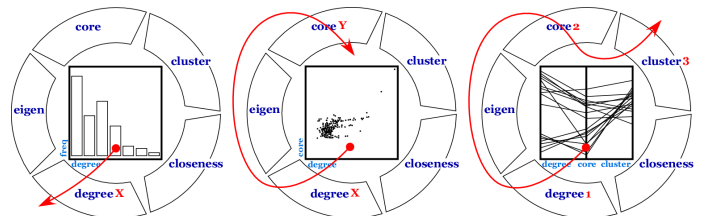


Fig. 4. Mock-up of alternative design: selecting a single dimension causes a histogram to be displayed (*Left*), selecting a second dimension causes a scatterplot to be displayed (*Middle*), selecting additional dimensions transitions to a parallel coordinate plot (*Right*).

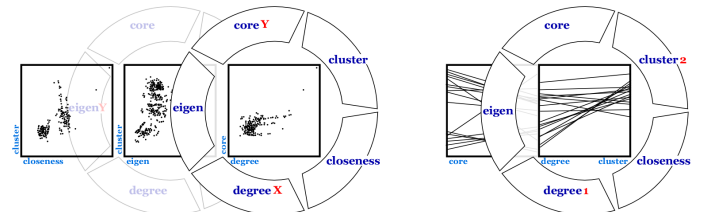


Fig. 5. Mock-up of a 2nd alternative design: Here, rather than “containing” a visualization, the FlowVizMenu is used to build up a visualization underneath, depositing or editing each piece of the visualization. *Left*: the FlowVizMenu is popped up over each cell of a matrix of scatterplots to choose the desired axes. *Right*: the FlowVizMenu is used to choose the axes within a parallel coordinate plot.

We hypothesized that because our implemented FlowVizMenu contains only a single scatterplot, it might be usable on a device with a small screen, and useful even without any additional views of the data. We have prototyped a FlowVizMenu with JavaScript that can be used on Apple’s iPhone and iPod touch devices (Figure 6). The size of the menu items and the surrounding space were chosen to reduce finger occlusion and to be able to move around the menu and scrub with ease. Such a widget might be combined with functionality in [6] for zooming in on the scatterplot.

In addition to the FlowVizMenu, users can also pop up a hotbox [25] to manually reposition nodes or manipulate the node-link diagram of the network.

5 ATTRIBUTE-DRIVEN LAYOUT

As already seen, the FlowVizMenu can be used to brush or select nodes that have, for example, both low degree but high betweenness centrality (such nodes could correspond to “bridges” in the network).

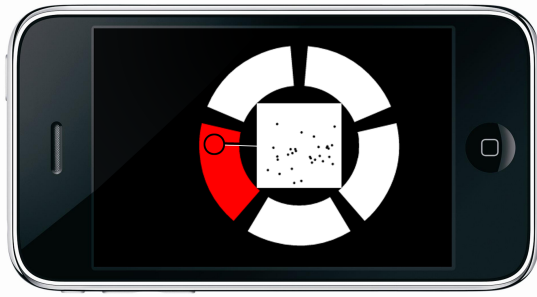


Fig. 6. Implemented version of the FlowVizMenu for iPhone/iPod touch platform. The dimensions within a single scatterplot can be transitioned with gestures.

Such functionality is useful when the node-link diagram is so dense that it is difficult to see individual nodes.

This section shows how the FlowVizMenu can also be used to drive an Attribute-Driven Layout (ADL) of the node-link diagram, or of a subset of nodes in the node-link diagram. For this, the user first activates the ADL scatterplot, which is displayed as a square region in the node-link diagram (see left sides of Figures 1 and 7). Then, when the user pops up the FlowVizMenu and selects two axes, the ADL scatterplot displays red and blue dots showing the corresponding scatterplot positions of selected and unselected nodes, respectively. Any nodes in the node-link diagram that have not been previously locked in place by the user are then animated to lie under their corresponding dots. The user may then select new axes within the FlowVizMenu, causing the positions of the red and blue dots to update immediately, after which the (unlocked) nodes again animate to their new positions under the dots. The use of smoothly animated transitions makes it easier to follow the trajectories of nodes and observe the behavior of groups of nodes during transitions.

As mentioned, with ADL, each unselected node is shown in black, and its corresponding position in the ADL scatterplot is shown with a blue dot. If the node is locked in some position different from the blue dot's position, the blue dot is easy to distinguish against the white background. However, if the node is not locked, then at the end of the animated transition the node moves under the blue dot, which then becomes difficult to see against the black background of the node. This was an intentional design choice, because the blue dot is almost redundant in that case. The red dots for selected nodes, however, are always visible, enhancing the visibility of selected nodes.

In between invocations of the FlowVizMenu, the user may also manually reposition nodes, lock or unlock certain nodes, and reposition unlocked nodes using force-directed layout. Because ADL only affects unlocked nodes, the user may freely mix manual layout, force-directed layout, and ADL. The result is a hybrid visualization that leverages the fact that scatterplots show two dimensions at once, while also allowing the user to see the edges (drawn as straight line segments) within the graph, and also mix in force-directed or manually repositioned nodes as desired. An alternative approach, where the user has two separate views (one of the node-link diagram and one scatterplot view), requires more screen space and could require more effort to mentally integrate the two views.

Nevertheless, the user may still sometimes prefer to deactivate the ADL scatterplot, so that they can keep the force-directed node-link diagram separate from, say, the 2D scatterplot in the FlowVizMenu (which allows for brushing and linking with the node-link diagram). The user is free to choose the approach best suited to their needs.

6 SCATTERPLOT MATRICES (SPLOMS)

The FlowVizMenu only displays a single scatterplot at a time, which can be useful for saving screen space. The user can also transition between different scatterplots within the FlowVizMenu with fast gestures. However, at times the user may want a more global view of

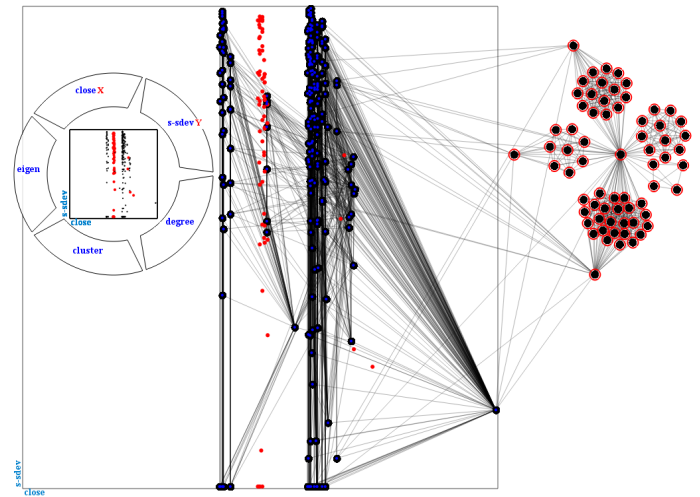


Fig. 7. A mixture of attribute-driven layout, force-directed layout, and manual layout. At right, four nodes were positioned manually and locked in place. Next, several of their neighbours (also shown at right) were positioned with force-directed layout and then locked in place. The remaining nodes, which were not locked by the user, are positioned according to the ADL scatterplot (left) being manipulated via the FlowVizMenu.

the data, and wish to compare several scatterplots simultaneously. For this, we use a scatterplot matrix (SPLOM).

The SPLOM and node-link diagram in our system can be zoomed and panned independently, and the FlowVizMenu's corresponding scatterplot is highlighted in the SPLOM. Brushing and linking can be performed between points in the SPLOM and nodes in the node-link diagram. Furthermore, the user can reduce the SPLOM to a single row of scatterplots, or even a single scatterplot, that can then be zoomed in to a larger size.

Our SPLOM can be ordered in the standard format shown in Figure 8 where each row contains only one kind of vertical axis, and each column contains only one kind of horizontal axis. Naturally, this simplifies comparison of scatterplots within the same row or column. Certain systems, such as [13], allow the rows and columns of a SPLOM to be reordered, for added flexibility, while still maintaining a single vertical axis within each row and a single horizontal axis within each column. Our system dispenses with this constraint, allowing other arrangements to be investigated. For example, the user may not be interested in comparing scatterplots with a common axis, or in seeing all possible scatterplots. Thus, we developed variants of the standard SPLOM, with different orderings of scatterplots or of their axes, which we now describe.

6.1 Ranked Scatterplot Matrix

Our first variant of the SPLOM orders scatterplots according to some ranking or metric of interest. Currently, our implementation of the ranked SPLOM orders scatterplots in descending order of the absolute value of their Pearson correlation coefficient, filling the first row from left-to-right, then the 2nd row, etc. This allows the user to see the scatterplots showing the strongest correlations at the top left of the matrix. It would also be possible to rank scatterplots according to any of the metrics listed in [36], or with the "quality-measures" in [1], or even allow the user to define their own "ranking" by dragging and dropping scatterplots to insert them at any position in the matrix.

6.2 Scatterplot Staircase (SPLOS)

This variant, which we call the Scatterplot Staircase or SPLOS (Figures 9A and 9C), shows only the scatterplots of consecutive pairs of dimensions, arranged in a staircase pattern, such that adjacent scatterplots share an axis along their common edge. To our knowledge, this staircase pattern of scatterplots is novel, but we note that it was inspired by quilts [33] (which use a staircase of adjacency matrices)

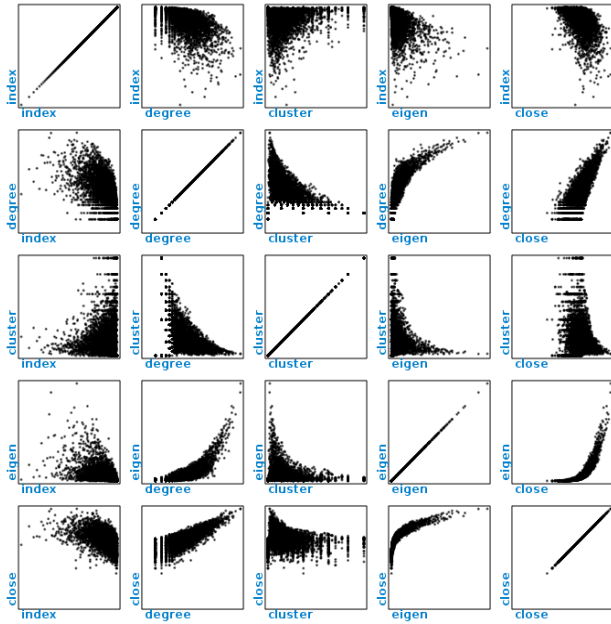


Fig. 8. A standard SPLOM (scatterplot matrix).

and also by an arrangement of adjacent scatterplots sharing axes on page 135 of [32]. Notice that the scatterplots involved in the SPLOS are located adjacent to the diagonal in the standard SPLOM. If the N dimensions in the data set are x_1, x_2, \dots, x_N , then the $N - 1$ scatterplots in the staircase are $x_2 \times x_1, x_2 \times x_3, x_4 \times x_3, x_4 \times x_5, \dots$. The entire staircase pattern is $\lfloor N/2 \rfloor$ columns wide and $\lceil N/2 \rceil$ rows tall. Although not implemented, it would be possible to rotate the staircase 45 degrees (Figures 9D, 10C) to take up less screen space. If each axis has length L , we can compare the space efficiency of the SPLOS, rotated SPLOS, and other multidimensional visualization techniques:

	Area Required	Aspect Ratio
Standard SPLOM (lower triangular half only)	$(N - 1)L \times (N - 1)L = \Theta(N^2L^2)$	1
Single Row of Scatterplots (e.g., Figures 10A, 10B, 11A)	$NL \times L = NL^2$	N
Scatterplot Staircase (SPLOS) (e.g., Figures 9A, 9C)	$\lfloor N/2 \rfloor L \times \lceil N/2 \rceil L = \Theta(\frac{1}{4}N^2L^2)$	$\Theta(1)$
Scatterplot Staircase (SPLOS) rotated 45° (e.g., Figures 9D, 10C)	$\frac{1}{\sqrt{2}}NL \times \frac{3}{\sqrt{2}}L = \frac{3}{2}NL^2$	$\frac{1}{3}N$
Parallel Coordinate Plot (PCP) (e.g., Figures 10D, 11D)	$(N - 1)kL \times L = \Theta(kNL^2)$	$\Theta(kN)$

where the Parallel Coordinate Plot (PCP) is assumed to have spacing between axes that is proportional to L (i.e., the spacing is kL , for some constant k), to prevent the slopes of line segments from becoming too extreme.

From the comparison, we see that the area required by the SPLOS is about 1/4 that required by a standard SPLOM. This means the user can have an overview of data that takes up less screen space, or that can be enlarged to take up the same space as a standard SPLOM but with each scatterplot 4 times larger. The area required by the rotated SPLOS is even smaller at $\frac{3}{2}NL^2$ (i.e., linear in N , and comparable to PCP). Comparing the rotated SPLOS with a single row of scatterplots, the rotated SPLOS requires 50% more area but has an aspect ratio that is 3 times closer to 1 (which may be advantageous in certain contexts) and also has the advantage that each dimension in the SPLOS lies on an axis shared by two scatterplots, easing comparisons. Note that the scatterplots used in the SPLOS are the same as those embedded in PCPs in [19], but with the advantage that pairs of scatterplots with a common axis are adjacent and aligned, again easing comparisons.

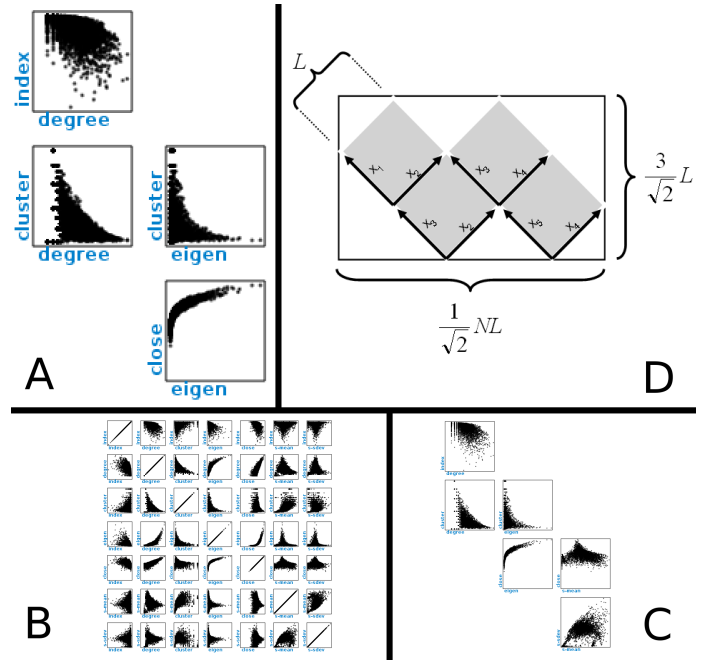


Fig. 9. *A*: a SPLOS of 4 scatterplots, for the same dataset and same 5 dimensions as in Figure 8. *B* and *C*: a standard SPLOM (*B*) of 7 dimensions, and the corresponding SPLOS (*C*) for the same 7 dimensions but composed of only 6 scatterplots. *D*: If there are N dimensions, and the $N - 1$ scatterplots have axes of length L , and the staircase is rotated 45° counterclockwise, the area of the bounding rectangle is linear in N .

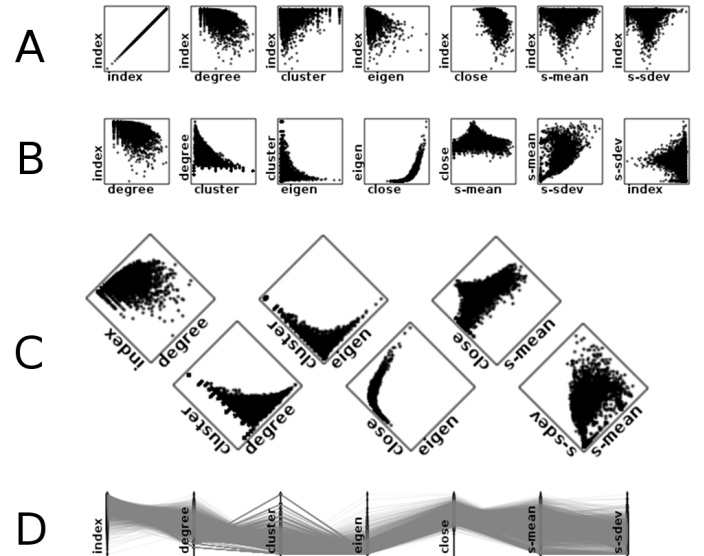


Fig. 10. Four visualizations of the same 7-dimensional network data. *A*: a single row of scatterplots from a standard SPLOM. Notice that one dimension (`index`) is crossed with all other dimensions. *B*: another row of scatterplots, now where each scatterplot involves 2 consecutive dimensions. *C*: a Scatterplot Staircase (SPLOS) rotated 45 degrees. Each adjacent pair of scatterplots share an axis. *D*: parallel coordinates plot (PCP). The SPLOS is scaled and positioned such that, for each axis shared by two scatterplots in the SPLOS, the midpoint of the axis is aligned with the corresponding axis of the PCP.

Finally, given the perceptual advantages of scatterplots over parallel coordinates [24], we feel the SPLOS is an interesting compromise between the standard SPLOM and PCPs.

6.3 Parallel Scatterplot Matrix (P-SPLOM)

Our system allows the scatterplots within a SPLOM to be rotated in 3D around either their vertical or horizontal axis. Figure 11 illustrates this for a single row of a SPLOM. Figures 11A through 11D show how a rotation around the vertical axes causes the visualization to transition from a sequence of scatterplots to a PCP. In between these two extremes, the user may rotate around both the vertical and horizontal axes (Figures 11E, F) yielding a 3D parallel coordinate plot (3D PCP) similar to those in [14, 28]. Although this is done for a single row of the SPLOM in Figure 11, such rotation is also allowed within the full matrix (Figures 12, 13). Because these visualizations combine SPLOMs, PCPs, and 3D PCPs with seamless transitions, we call this combination based on rotation a Parallel Scatterplot Matrix (P-SPLOM).

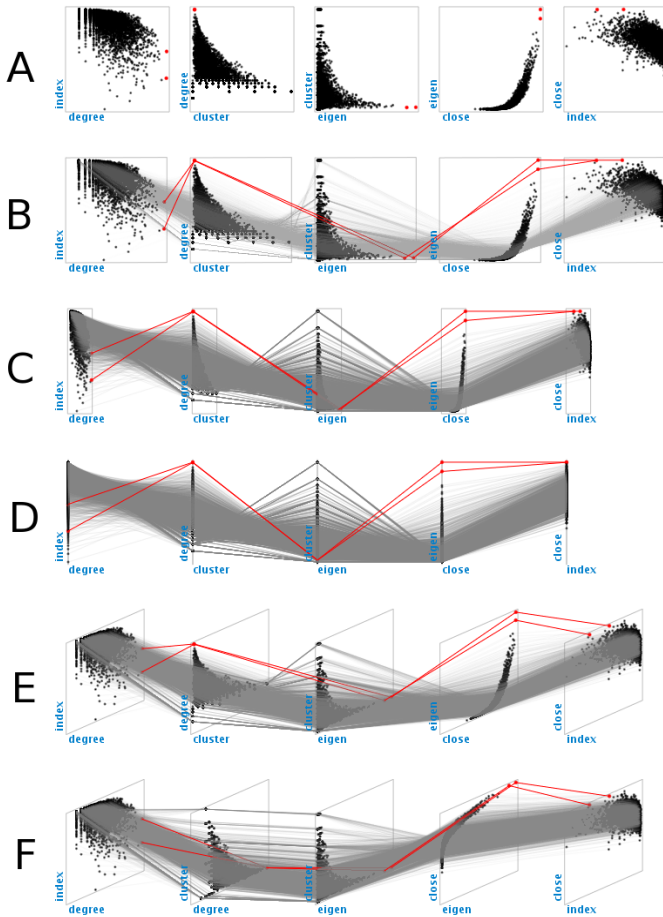


Fig. 11. A single row of a P-SPLOM. *A*: a single row of scatterplots, with two points selected and highlighted in red. *B-C*: as the scatterplots are rotated around their vertical axes, links (in grey) between corresponding points are faded in. *D*: after rotating 90 degrees, the result is a Parallel Coordinate Plot (PCP). *E*: during rotation, the user may also rotate around the horizontal axes, resulting in a 3D PCP. *F*: an alternate ordering of axes for 3D PCP, facilitating comparison between consecutive pairs of scatterplots.

Several orderings of the axes are possible within P-SPLOMs. For illustration purposes, consider first the axes within a standard SPLOM (Figure 8). If there are five dimensions named A, B, C, D, and E, we could specify the axes within the standard SPLOM with the following table:

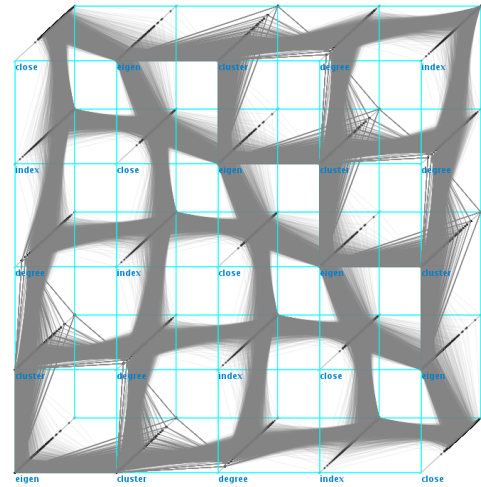


Fig. 12. Rotation of scatterplots within a P-SPLOM is possible along the horizontal and vertical axes simultaneously. Here, the user has rotated the scatterplots approximately 90 degrees around the horizontal axes, and 45 degrees around the vertical axes. Links between corresponding points are shown in grey. Each row and each column of this matrix could be further rotated into a PCP.

Standard SPLOM									
horizontal axes					vertical axes				
A	B	C	D	E	A	A	A	A	A
A	B	C	D	E	B	B	B	B	B
A	B	C	D	E	C	C	C	C	C
A	B	C	D	E	D	D	D	D	D
A	B	C	D	E	E	E	E	E	E

If the above ordering of axes is used within a P-SPLOM, the PCPs that result after rotation are not useful, because each row and each column will contain the same PCP axis repeated five times (see Figure 13, top row). Thus, our system also allows an ordering of axes that we call *Doubly-Latin*, because it involves two Latin squares:

Doubly-Latin SPLOM									
horizontal axes					vertical axes				
E	D	C	B	A	A	B	C	D	E
A	E	D	C	B	B	C	D	E	A
B	A	E	D	C	C	D	E	A	B
C	B	A	E	D	D	E	A	B	C
D	C	B	A	E	E	A	B	C	D

An example of this ordering is in Figure 13 (2nd row). The Doubly-Latin ordering is useful for transitioning to PCPs because every row contains each kind of vertical axis once, and every column contains each kind of horizontal axis once. Thus, if the user focuses on any single row or column and rotates toward a PCP, all dimensions will be visible after the rotation within that one row or column. Figure 12 shows the full matrix for a Doubly-Latin P-SPLOM during rotation. Each column and each row of this figure is a 3D PCP that would (after complete rotation) contain all axes. As a side note, if the number of dimensions is odd, the Doubly-Latin ordering results in an Euler square (also known as a Graeco-Latin square), which has the property that every possible pair of dimensions occurs once. This is useful *before* rotating toward PCPs, since it guarantees all possible scatterplots will be visible in the full matrix. In other words, the Euler square is a permutation of the scatterplots in the standard SPLOM.

At the same time, the Doubly-Latin ordering has the disadvantage that within any given row or column, there can be redundant scatterplots. For example, within the Doubly-Latin matrix shown in Figure 13, the first row of scatterplots is $E \times A, D \times B, C \times C, B \times D, A \times E$; i.e., the last two scatterplots are transpositions of the first two. The same redundancy is seen in the Doubly-Latin scatterplots in Figure 14.

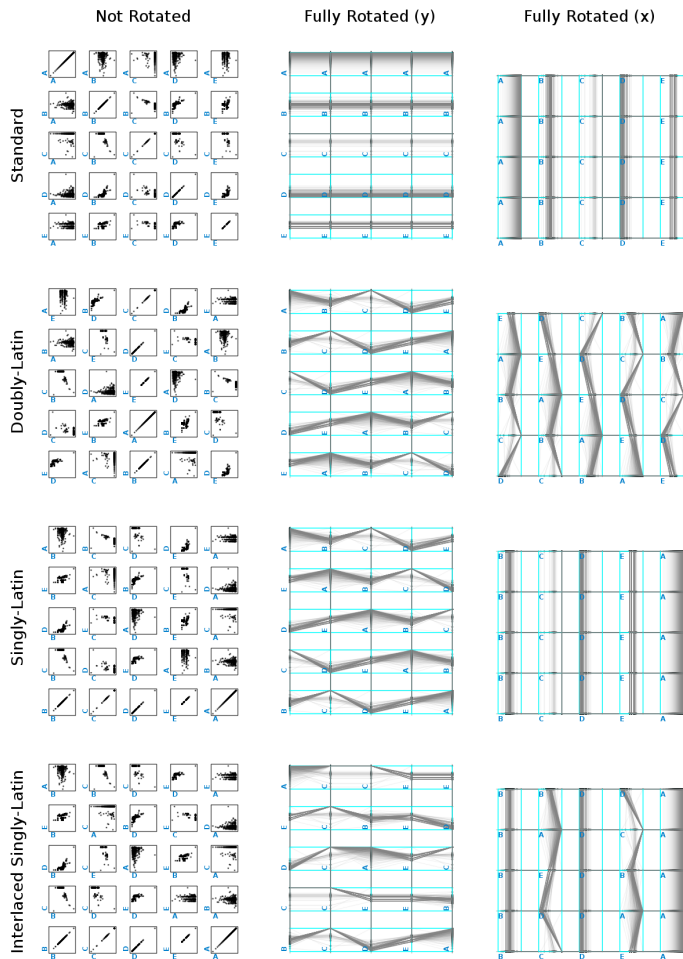


Fig. 13. Each row shows a different ordering of the dimensions within the P-SPLOM. The left column shows the full scatterplot matrix (without rotation), the middle column shows the PCP resulting after a 90 degree rotation around the y axes, and the right column shows the PCP resulting after a 90 degree rotation around the x axes.

This redundancy does not matter after a full rotation toward PCPs, but it does make the Doubly-Latin ordering less attractive for use with 3D PCPs of a single row or column. Thus, our system also allows for what we call a *Singly-Latin* ordering, because it only involves one Latin square:

Singly-Latin SPLOM									
horizontal axes					vertical axes				
B	C	D	E	A	A	B	C	D	E
B	C	D	E	A	E	A	B	C	D
B	C	D	E	A	D	E	A	B	C
B	C	D	E	A	C	D	E	A	B
B	C	D	E	A	B	C	D	E	A

The above Singly-Latin ordering has the advantage that within any given row (but not column), we find each vertical axis once (making it good for rotation toward PCP) and each horizontal axis once. Furthermore, within any given row, there are no redundant scatterplots, making the ordering reasonable for 3D PCPs. For example, Figures 11A through 11E show the first row of the Singly-Latin P-SPLOM whose ordering is given above. Notice the pattern of pairs of dimensions in the 1st row: $B \times A, C \times B, D \times C, \dots$ (i.e., the x axis of one is the y axis of the next). The same pattern is seen in Figure 10B.

There is, however, still a minor disadvantage of the Singly-Latin ordering for 3D PCPs: within the 1st row, each pair of consecutive scatterplots share a dimension on *different* axes, making comparisons more

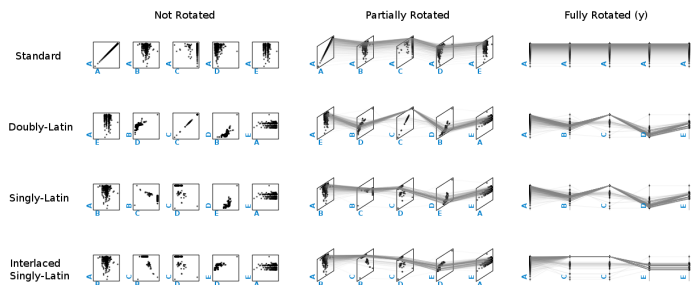


Fig. 14. Each row shows a different ordering of the dimensions within a single row of scatterplots. The left column shows the scatterplots without rotation, the middle column shows the 3D PCP resulting after a partial rotation, and the right column shows the PCP resulting after a 90 degree rotation around the y axes.

difficult. For example, in Figure 11E, the left-most scatterplot has *degree* on its horizontal axis, and the next scatterplot has *degree* on its vertical axis. Thus, our system also supports an *Interlaced Singly-Latin* ordering, which results from taking the Singly-Latin ordering and swapping the axes of scatterplots in every other column. The first row of an Interlaced Singly-Latin P-SPLOM is shown in Figure 11F. Notice now that *degree* is the horizontal axis of the 1st and 2nd scatterplots, and *cluster* is the vertical axis of the 2nd and 3rd scatterplots, etc. Notice also that the scatterplots in Figure 11F are the same as in the staircase pattern of Figure 9A! This ordering facilitates comparisons across the scatterplots when using the 3D PCP.

The following table summarizes the tradeoffs between the different orderings for P-SPLOMs:

	full scatterplot matrix (not rotated) Fig. 13	single row of scatterplots (not rotated) Fig. 14	single row 3D PCP (partially rotated) Fig. 14	single row PCP (fully rotated) Fig. 14
Standard	best	good	good	worst
Doubly-Latin	redundant if N even	redundant	redundant	best
Singly-Latin	good	good	good	best
Interlaced Singly-Latin	good	good	best	redundant

The user may switch between all these orderings at run time, triggering a smoothly animated transition from the old ordering to the new ordering. The user may also switch between viewing the full matrix or focus on a single row, and may also transition between “flat” scatterplots to 3D PCP or PCP through rotation (Figure 11). Thus, the user is free to choose the visualization that best suits their needs. Nevertheless, the above table indicates that the Singly-Latin ordering may be the best compromise across all the views of the data, since it is the only one that avoids redundancy in all cases.

7 INITIAL USER FEEDBACK

We asked five bioinformaticians (four men and one woman, between 24 and 43 years old) to help us evaluate our system’s user interface for the exploration of biological networks. All five participants are experienced computer users and work with network data.

Each participant was given a ten minute demonstration and explanation of the interface, in part to demonstrate some of the possibilities afforded by metric-based exploration. Next, a set of questions were asked regarding the potential relevance of metric-based graph exploration for bioinformatics. Participants were then allowed to freely explore the interface while thinking aloud. A second set of open-ended questions was then asked regarding the user interface and the participant’s impressions.

Participants quickly understood the logic behind each interface element. The effectiveness of the interface for selection was noted in particular (“it’s a very quick way to pick out interesting outliers”, “it allows you to swiftly do something that might have taken a lot longer just manually picking through the graph.”)

Smoothly animated transitions were seen as useful (“The transition is helpful, because I think if it was just done without transition, then you would lose sight of [the data]”). Four participants indicated they had already worked with most of the metrics available in the interface. However, the ability to see these metrics plotted against each other prompted new questions (“it’s maybe not as intuitive to me what the correlation between, let’s say, degree and betweenness centrality, which I know is quite correlated right, but what’s the meaning of that?”, “What an outlier would mean there, I’d like to know, to have a better sense of that.”) All participants saw the potential for making new kinds of biological queries, even if it was initially unclear what kind of queries would be best.

One participant stated “It would be cool if there were some sort of protein, say, that had high degree and low betweenness centrality, some unexpected relationship.” Upon hearing this, one of us (Jurisica) pointed out that the opposite would be interesting: a node with high centrality and low degree would correspond to a node that, if removed, would disconnect the graph.

All participants stated they would use the interface in their daily work if it were available to them.

8 EXAMPLE USE WITH REAL-WORLD DATA

To further evaluate our prototype, we tested its use with a real-world biological network (Figure 15). We wanted to visually explore the network and see how we could make use of the advanced features of the interface, going beyond simply selecting clusters or high degree nodes as is possible with status quo software. First, in the P-SPLOM and in the force-directed layout, we could see that the network was composed of a few nodes of high degree surrounded by many nodes of degree one. A preliminary layout was made by combining FlowVizMenu selection with hotbox [25] layout commands. The highest-degree nodes were selected using the FlowVizMenu, and then laid out in a circular pattern (and locked in place) using the hotbox. The degree-one neighbors of the highest-degree nodes were then selected and laid out on a secondary circle (also locked in place) surrounding the first. The remaining nodes, in the center of these two circles, were structurally less obvious. To find interesting features in these remaining nodes, we tried to follow the insight that a node with low degree but high centrality could be a good candidate to investigate. The FlowVizMenu was used to select these two dimensions and steer the ADL view. One node stood apart and corresponded to our criteria. A direct selection in the ADL revealed the name of one particular protein, which our bioinformatics collaborators intend to further investigate.

9 CONCLUSIONS AND FUTURE DIRECTIONS

We have presented novel approaches to multivariate graph visualization that integrate previous techniques, through popup gestural interaction (the FlowVizMenu) and through the use of hybrids (the P-SPLOM which unifies scatterplot matrices, normal 2D parallel coordinates, and 3D parallel coordinates, with seamless transitions between them; and our attribute-driven layout which can be mixed with force-directed layout and manual positioning). We have also presented an investigation of possible orderings of dimensions within P-SPLOMs, and the novel Scatterplot Staircase (SPLOS). Together, we feel these techniques provide a flexible toolbox for dissecting a network, isolating nodes of interest, and manipulating the layout. We also note that the P-SPLOM and SPLOS are general multidimensional visualization techniques that could be applied to non-network data.

Possible future directions include: deploying the techniques as a plugin for biological network software such as NAViGaTOR [5] or Cytoscape; applying the FlowVizMenu, P-SPLOM and SPLOS to non-network data; and further adapting the FlowVizMenu or related techniques for use on small screens or mobile platforms.

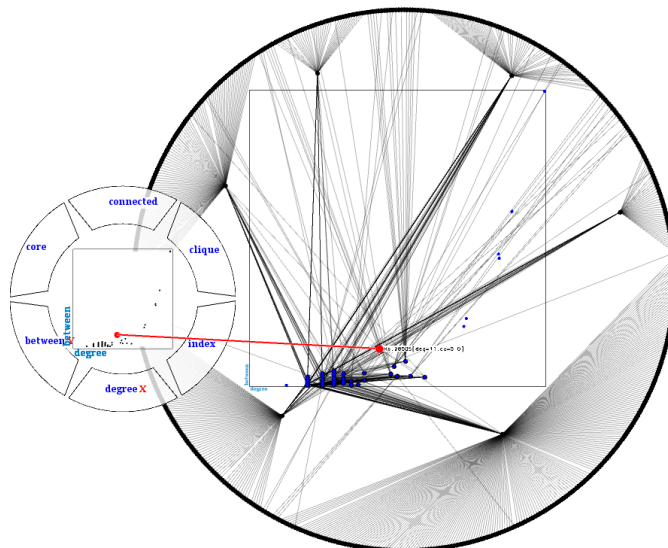


Fig. 15. Example use of the FlowVizMenu combined with ADL to explore a biological network. Prior to using the ADL shown here, the highest-degree nodes and their low-degree neighbors were manually positioned onto circular layouts. The remaining nodes are laid out with ADL to find a candidate node with high centrality and low degree, since such properties could signify it acts as a bridge, having special biological significance.

One question that remains open is what value, if any, there is in displaying a full matrix of parallel coordinates (Figures 12 and 13). Currently, there is much redundancy in these views due to pairs of axes being repeated. However, a different ordering of dimensions might eliminate this redundancy. Otherwise, users may be best to reduce the matrix to a single row or column before rotating to PCP.

There are also several avenues open for future evaluation. For example, how does user performance with our FlowVizMenu compare to that with GraphDice’s control menu? How accepting are users of SPLOMs that do not use the standard ordering? These can only be answered by future work.

ACKNOWLEDGMENTS

We thank our collaborators at SAP Business Objects, members of Jurisica Lab at OCI, and members of the Multimedia Lab at ETS for their feedback. Pascal Viau helped program the iPod touch version of the FlowVizMenu. This research was funded by an SAP Business Objects ARC Fellowship, NSERC, and the FQRNT.

REFERENCES

- [1] G. Albuquerque, M. Eisemann, D. J. Lehmann, H. Theisel, and M. Magnor. Quality-based visualization matrices. In *Proceedings of Vision, Modeling, and Visualization (VMV)*, 2009.
- [2] A. Aris and B. Shneiderman. Designing semantic substrates for visual network exploration. *Information Visualization*, 2007.
- [3] D. Auber, Y. Chiricota, F. Jourdan, and G. Melançon. Multiscale visualization of small world networks. In *Proceedings of IEEE Symposium on Information Visualization (InfoVis)*, 2003.
- [4] A. Bezerianos, F. Chevalier, P. Dragicvic, N. Elmqvist, and J.-D. Fekete. GraphDice: A system for exploring multivariate social networks. In *Proceedings of Eurographics/IEEE-VGTC Symposium on Visualization (EuroVis)*, 2010.
- [5] K. R. Brown, D. Otasek, M. Ali, M. J. McGuffin, W. Xie, B. Devani, I. L. van Toch, and I. Jurisica. NAViGaTOR: Network analysis, visualization and graphing Toronto. *Bioinformatics*, 25(24):3327–3329, 2009.
- [6] T. Büring, J. Gerken, and H. Reiterer. User interaction with scatterplots on small screens - a comparative evaluation of geometric-semantic zoom and fisheye distortion. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 12(5):829–836, 2006.

- [7] W. Buxton. Chunking and phrasing and the design of human-computer dialogues. In *Proc. IFIP World Computer Congress*, pages 475–480, 1986.
- [8] J. Callahan, D. Hopkins, M. Weiser, and B. Shneiderman. An empirical comparison of pie vs. linear menus. In *Proc. ACM Conference on Human Factors in Computing Systems (CHI)*, pages 95–100, 1988.
- [9] Y. Chiricota, F. Jourdan, and G. Melançon. Metric-based network exploration and multiscale scatterplot. In *Proceedings of IEEE Symposium on Information Visualization (InfoVis)*, pages 135–142, 2004.
- [10] C. Collins and S. Carpendale. VisLink: Revealing relationships amongst visualizations. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 13(6):1192–1199, 2007.
- [11] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, 1999.
- [12] M. d’Ocagne. *Coordonnées parallèles et axiales: méthode de transformation géométrique et procédé nouveau de calcul graphique déduits de la considération des coordonnées parallèles*. Gauthier-Villars, 1885.
- [13] N. Elmqvist, P. Dragicevic, and J.-D. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 14(6):1141–1148, 2008.
- [14] G. Falkman. Information visualisation in clinical odontology: multidimensional analysis and interactive data exploration. *Artificial Intelligence in Medicine*, 22:133–158, 2001.
- [15] E. Fanea, S. Carpendale, and T. Isenberg. An interactive 3D integration of parallel coordinates and star glyphs. In *Proceedings of IEEE Symposium on Information Visualization (InfoVis)*, pages 149–156, 2005.
- [16] F. Guimbretière. Measuring flowmenu performance. Technical Report CS-TR-4408, UMIACS-TR-2002-88, HCIL-TR-2002-17, University of Maryland, January 2003. <http://hdl.handle.net/1903/1232>.
- [17] F. Guimbretière and T. Winograd. FlowMenu: Combining command, text, and data entry. In *Proc. ACM UIST*, pages 213–216, 2000.
- [18] J. A. Hartigan. Printer graphics for clustering. *Journal of Statistical Computation and Simulation*, 4(3):187–213, 1975.
- [19] D. Holten and J. J. van Wijk. Evaluation of cluster identification performance for different pcv variants. In *Proceedings of Eurographics/IEEE-VGTC Symposium on Visualization (EuroVis)*, 2010.
- [20] A. Inselberg. The plane with parallel coordinates. *Visual Computer*, 1:69–91, 1985.
- [21] J. Johansson, P. Ljung, M. Jern, and M. Cooper. Revealing structure in visualizations of dense 2D and 3D parallel coordinates. *Information Visualization*, 5:125–136, 2006.
- [22] G. Kurtenbach and W. Buxton. The limits of expert performance using hierarchic marking menus. In *Proc. ACM CHI*, pages 482–487, 1993.
- [23] G. Kurtenbach, G. Fitzmaurice, R. Owen, and T. Baudel. The Hotbox: Efficient access to a large number of menu-items. In *Proc. ACM CHI*, 1999.
- [24] J. Li, J.-B. Martens, and J. J. van Wijk. Judging correlation from scatterplots and parallel coordinate plots. *Information Visualization*, 9:13–30, 2010.
- [25] M. J. McGuffin and I. Jurisica. Interaction techniques for selecting and manipulating subgraphs in network visualizations. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 15(6):937–944, 2009.
- [26] S. Pook, E. Lecolinet, G. Vaysseix, and E. Barillot. Control menus: Execution and control in a single interactor. In *Extended abstracts of CHI*, 2000.
- [27] H. Qu, W.-Y. Chan, A. Xu, K.-L. Chung, K.-H. Lau, and P. Guo. Visual analysis of the air pollution problem in Hong Kong. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 13(6):1408–1415, 2007.
- [28] O. Rübél and et al. PointCloudXplore: Visual analysis of 3D gene expression data using physical views and parallel coordinates. In *Proceedings of Eurographics/IEEE-VGTC Symposium on Visualization (EuroVis)*, 2006.
- [29] A. J. Sellen, G. P. Kurtenbach, and W. A. S. Buxton. The prevention of mode errors through sensory feedback. *Human Computer Interaction*, 7(2), 1992.
- [30] C. A. Steed, J. E. Swan II, T. J. Jankun-Kelly, and P. J. Fitzpatrick. Guided analysis of hurricane trends using statistical processes integrated with interactive parallel coordinates. In *Proc. IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 19–26, 2009.
- [31] S. T. Teoh, K.-L. Ma, S. F. Wu, and T. J. Jankun-Kelly. Detecting flaws and intruders with visual data analysis. *IEEE Computer Graphics and Applications (CG&A)*, 24(5):27–35, 2004.
- [32] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 1983.
- [33] B. A. Watson, D. Brink, M. Stallmann, R. Devajaran, M. Rakow, T.-M. Rhyne, and H. Patel. Matrix depictions for large layered graphs. Technical Report TR-2008-17, Dept. Computer Science, North Carolina State University, 2008. <http://designgraphics.ncsu.edu/pages/projects/quilts/>.
- [34] M. Wattenberg. Visual exploration of multivariate graphs. In *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI)*, pages 811–819, 2006.
- [35] E. J. Wegman. Hyperdimensional data analysis using parallel coordinates. *J. of the American Statistical Association*, 85(411):664–675, 1990.
- [36] L. Wilkinson, A. Anand, and R. Grossman. Graph-theoretic scagnostics. In *Proc. IEEE Symposium on Information Visualization (InfoVis)*, 2005.
- [37] S. Wuchty and E. Almaas. Peeling the yeast protein network. *Proteomics*, 5:444–449, 2005.
- [38] Y. Xu, W. Hong, X. Li, and J. Song. Parallel dual visualization of multidimensional multivariate data. In *Proceedings of IEEE International Conference on Integration Technology*, pages 263–268, 2007.
- [39] X. Yuan, P. Guo, H. Xiao, H. Zhou, and H. Qu. Scattering points in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 15(6):1001–1008, 2009.