MatLink: Enhanced Matrix Visualization for Analyzing Social Networks

Nathalie Henry, Jean-Daniel Fekete

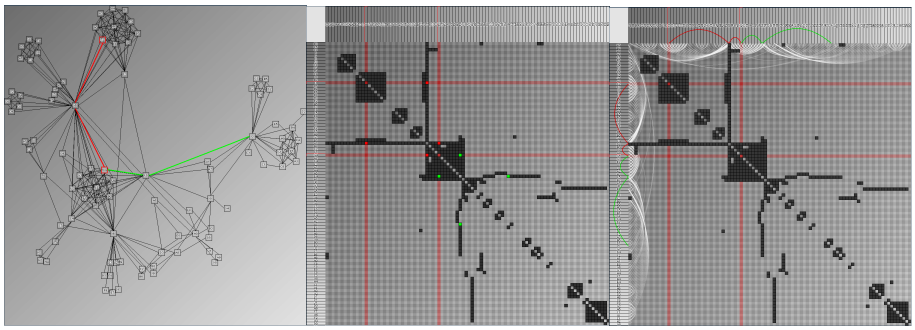
MatLink: Enhanced Matrix Visualization for Analyzing Social Networks

Nathalie Henry[1,2] and Jean-Daniel Fekete[1]

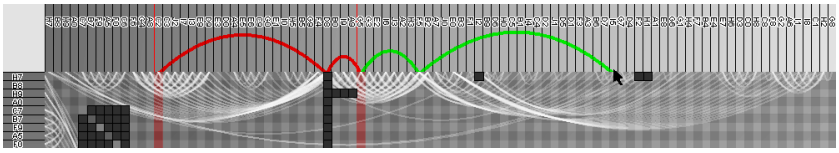
[1] INRIA/LRI, Bat. 490, Univ. Paris-Sud, F91405, Orsay, France
[2] Univ. of Sydney, Australia

Abstract. Visualizing social networks presents challeges for both node-link and adjacency matrix representations. Social networks are locally dense, which makes node-link displays unreadable. Yet, main analysis tasks require following paths, which is difficult on matrices. This article presents MatLink, a hybrid representation with links overlaid on the borders of a matrix and dynamic topological feedback as the pointer moves. We evaluated MatLink by an experiment comparing its readability, in term of errors and time, for social network-related tasks to the other conventional representations on graphs varying in size (small and medium) and density. It showed significant advantages for most tasks, especially path-related ones where standard matrices are weak.

Key words: Node-Link Diagram, Matrix Visualization, Social Network.



(a) Node-Link(NL) (b) Matrix(MAT) (c) MatLink



(d) Zoom on MatLink

Fig. 1: Three representations of a social network.

1 Introduction

Social network analysis has been a growing area of the social sciences recently for many reasons: Internet social activities that can be automatically instrumented and analyzed: Peer-to-Peer file exchanges, chat, blogs, and collective development such as Wikipedia and Open-Source Software projects; intelligence agencies seeking to discover terrorist networks; monitoring to detect and contain outbreaks of diseases such as avian influenza and SARS.

Many of these new social networks are large, complex and continuously changing, which creates challenges for analysis tools. Information visualization can be an effective approach to help social science researchers both explore relationships between actors and present their findings to others. For both purposes, it is critical to make the visual representations of social networks *readable*.

Examination of the "International Network for Social Network Analysis" software repository (`http://www.insna.org`) reveals that most of the 55 software referenced for visualizing and exploring social networks use Node-Link diagrams (NL). Unfortunately, social networks are locally dense, and NL have been shown by Ghoniem et al. [1] to behave poorly on dense networks even for the simplest tasks such as checking if one node is connected to another. NL remain useful to show the overall structure of a network (Figure 1a), but details about dense sub-graphs within them are frequently impossible to read. [1] also showed that matrix graph representations (MAT) (Figure 1b), the primary alternative to NL, are poor for path-finding tasks. These are crucial to social network analysis [2].

To address these limitations, we developed MatLink, an enhanced matrix-based graph visualization that overlays a linear node-link diagram on the edges and adds dynamic feedback of relationship between nodes (Figure 1c). To assess its effectiveness, we performed an experiment comparing user performance with NL, MAT and MatLink on a set of representative social network analysis tasks.

The rest of the paper is organized as follows: the next section describes previous work on social network characterization, evaluating graph visualizations and analysis tasks, visual exploration systems and layout algorithms. We then describe our novel visualization and present an experimental evaluation comparing it to existing network visualizations. A discussion on the results and their implications follows.

2 Related Work

2.1 Social Network Characterization

Social networks involve persons or groups called actors and relationships between them, with a lot of variety in the kind of actors and relationships. As described in Wasserman and Faust [2], actors can be people, subgroups, organizations or collectivities; relations may be friendship (relationships), interactions, communications, transactions, movement or kinship. However, the nature of actors and relations does not really matter: we focus on their structure.

Very often in the literature, social networks are confused with small-world networks. After studying real social networks, we identified three categories.

Almost trees are trees with additional links forming cycles with a low probability. They include genealogy data and Sexually-Transmitted Disease transmission patterns.

Almost complete graphs are complete graphs with missing relations, such as data about trade between countries, cities or companies. They usually carry values on their edges.

Small-world networks have been studied extensively since they were first described in Watts and Strogatz [3], who defined them as graphs with three properties: power-law degree distribution, high clustering coefficient and small average shortest path. They are locally dense (sparse with dense sub-graphs.)

Ghoniem at al. [1] showed that NL are very poor at visualizing almost complete graphs and more generally dense networks. Conversely, NL are very well suited to visualizing "almost trees" networks. Since Small-World networks are locally dense, NL readability is questionable.

2.2 Evaluating Graph Visualizations and Task Taxonomies

Selecting representative and comparable tasks to compare visualization performance is difficult. Most early visualization system evaluators chose tasks they considered representative and that they knew their system could perform. However, evaluating multiple systems calls for a more systematic approach.

Task taxonomies have been proposed for many kinds of visualizations. Amar et al. presented a list of 10 basic tasks on multivariate data visualizations [4] Plaisant et al. adapted it to graphs [5]. Unlike multivariate data tables where the objects of interest are records with uniform attribute sets, networks have attributes associated with edges as well as vertices, and a more complex topology including compound objects such as paths or sub-graphs. Thus the task set is different. However, their tasks are generic and need extension for social networks.

Reference books on social network analysis such as [2] list the following important high-level concepts: *centrality*, *cohesive subgroups*, *positions* and *roles*. Each of these concepts are formalized with several measures, all of them requiring computations on the network. However, visually, they all rely on path-related properties. For example, centrality measures include *betweenness centrality*: the most central being the person who is on most shortest-paths. Cohesive subgroups, positions and roles also rely on path-related measures. Therefore, we consider that a good representation of a social network should visually support path-related tasks.

Ghoniem et al. published an experimental evaluation comparing performance in NL and MAT, but only for low-level readability tasks [1]. We are not aware of comparisons targeted at social networks.

2.3 Systems for Social Network Exploration

We distinguish two categories of social network visualization systems: systems for end users to visualize their personal networks, and those designed for professionals analyzing entire networks.

Systems such as Vizster [6] or flickrGraphc (`http://www.marumushi.com/apps/flickrgraph`) are in the first category. They are an easy and entertaining means for users to explore their personal networks by interactive navigation. They do not provide an overview of the full network and only use NL.

Systems in the second category focus on professional analysts willing to spend significant time and effort to learn the system and analyze their data. Pajek [7], the most popular system among social science researchers, offers a sometimes-intimidating set of menus and functions to analyze large social networks. JUNG(`http://jung.sf.net`) also provide a rich set of drawing functions. Guess [8] takes a different approach, letting users write simple script-language programs. All these systems use only NL, except that Pajek can perform block modeling [9] and print (but not display) their adjacency matrices.

MatrixExplorer [10] uses NL and MAT representations in parallel to visualize and explore social networks. Users can perform each task on the most appropriate representation, maintaining a visual relation between them by interactive brushing and linking. However, alternating between two representations imposes a significant cognitive load. Also, MatrixExplorer requires at least two screens to be used effectively.

2.4 Layout for Node-Link and Matrix representations

Besides choosing an appropriate representation, its layout must be correct if it is to be readable, revealing the network's overall structure and important features. Layout algorithms compute positions for each vertex from the graph topology, sometimes augmented with a (dis)similarity function between vertices. For NL, a position is a 2D or 3D point; for MAT, a 1D ordering. Some layout methods are specific for one representation, while others can be applied to both. We review those considered most effective for social network visualizations.

Good introductions to NL layout algorithms can be found in [11]. For social networks, the biggest problem is that the distribution of edge numbers is not linear but power-law. Most vertices have few edges but few — often the most interesting ones — have many, with a very skewed distribution. Simple force-based graph layout algorithms behave poorly on such networks. Recently, Noack presented important improvements [12]. His layout behaves as well as possible on social networks. However, the dense sub-graphs around high-degree vertices remain difficult or impossible to read.

Several strategies have been investigated to overcome the link density problem: clustered graphs [13] in which dense graph parts are collapsed and PivotGraphs [14] where the graph is only visible according to the values of selected vertex attributes. Network details these solutions hide remain accessible by navigation or interactions.

Matrix representations have also been used to visualize social networks. There is a long tradition of matrix block modeling in the social sciences [9]. As Bertin stated, when adequately reordered, a matrix can reveal both global and local structures in a network [15]. While block modeling tries to gather equivalent roles into blocks, other reordering algorithms collect vertices with similar con-

nection patterns. Several methods attempt to gather clusters that exhibit structural features such as cliques or quasi-cliques, bridges or articulation points, communities and outliers. Henry and Fekete [10] compared several categories of methods. They showed that Traveling Salesman Problem (TSP) approximation and Clustering methods produce better results when applied to the distance matrix of the graph instead of its boolean adjacency matrix. A survey remains to be done to clearly understand the advantages and drawbacks of each method.

Matrices display dense networks without edge overlap, but have display area quadratic in the number of vertices. Techniques have been designed to navigate effectively on very large matrices. Abello and Van Ham [16] introduced matrices augmented with clustering trees. Van Ham [17] described smooth navigation techniques for matrices whose vertices possess several hierarchical levels.

3 MatLink

To address matrix-based graph representations' weaknesses on the path-related tasks social network analysis requires, we designed MatLink, a matrix representation with links overlaid on its borders and interactive drawing of additional links and highlighting of the cells included in a path from the cell under the mouse pointer.

MatLink displays the full graph using a linearized node-link representation we call the full linear graph (Figure 1d). Its links are curved lines drawn interior to the vertex displays at the top and left edges of the matrix. Links are drawn over the matrix cells, using transparency to avoid hiding them. Longer links are drawn above shorter ones. The linear graph conveys detailed and long-range structure together without hiding any detail of the matrix: a feeling for link densities and sub-graphs, but also paths and cut points.

When the user has selected a vertex in the rows or columns, it is highlighted in red, and the shortest path between this vertex and the one currently under the mouse pointer is drawn in green on the vertex area, mirror-imaged to the links drawn in the matrix border[3]. This dynamic visualization of the shortest path is designed to make paths preattentively visible on the matrix. Early versions of MatLink drew these dynamic paths over the full linear graph, but users complained about visual complexity and difficulty seeing cells under the path links. This was not the case with paths drawn in the vertex area.

When several vertices are selected, their related rows or columns are highlighted in red, and the shortest path is visualized using red curved links in the vertex area. When the user moves the mouse pointer, the shortest path between the last selection and the vertex under the pointer is drawn in green.

We display only one shortest path even if several equivalent ones exist. To avoid confusion, we ensure that the same path is always displayed between two vertices. If the path from A to E contains C, then the same links will be displayed when only the path between A and C is visible. We considered also showing a shortest cycle within a subset of vertices but decided it would be confusing,

[3] A video is available at `http://insitu.lri.fr/~nhenry/matlink/matlink.mov`

because adding one vertex to the selection could completely change the links drawn. Moreover, when the selection grows, computing a shortest cycle requires noticeable time. Other analytical attributes could be displayed on the linear graph, statically or dynamically. In this article, we focus on the shortest path.

Displaying the linear graph of a randomly permuted matrix is useless and even confusing. However, after the matrix is reordered using the modified TSP algorithm described in [10], the matrix representation shows clusters clearly. Also, the linear graph appears well organized with mostly short links connecting nearby vertices and some long edges connecting the clusters. Although our layout algorithm does not guarantee that the total number of crossings is minimized, it is very small compared with a random layout. This layout also reveals that some vertices primarily belonging to one cluster also have a few links to vertices in other clusters, sometimes not visible on the screen when the network is large and requires scrolling. Finally, the linear graph seems to facilitate the understanding of the matrix representation when users are familiar with NL.

MatLink is implemented with the InfoVis Toolkit [18], so all the attributes of the network can be assigned as visual attributes such as color, or label for the vertices and the edges.

4 Experimental Evaluation

We performed a controlled experiment to compare MatLink with MAT and NL. We designed a 3x6x5 within-subject experiment, comparing the 3 visualizations of 6 social networks asking subjects to complete 5 different tasks for each.

4.1 Selected tasks

This experiment evaluated primarily mid-level readability tasks. Low-level tasks have been evaluated by Ghoniem et al. [1] for MAT and NL (whose paradigms MatLink combines) while performance on high-level interpretation tasks depends more on the domain and the subject's background, requiring an subjective and time-consuming evaluation.

We selected the three most important high-level tasks in social network analysis: evaluating connectivity, finding central actors and identifying communities. We evaluated performance on one or two medium-level tasks within each of these high-level tasks. We also chose tasks we could easily explain to novice users, and for which answers could be objectively validated.

1. *commonNeighbor*: given two actors, find an actor directly linked to both;
2. *shortestPath*: given two actors, find a shortest path linking them;
3. *mostConnected*: find the actors with the highest number of relations;
4. *articulationPoint*: find a cut point, i.e. an actor linking two sub-graphs;
5. *largestClique*: find the largest set of actors who are all linked to each other.

4.2 Dataset selection

To avoid scrolling and navigation issues, we limited dataset sizes to what all three representations could display on one screen, slightly less than a hundred nodes. We also only considered undirected networks and used the largest connected component.

Because existing social network generators did not provide realistic data (for details, see `http://www.infovis-wiki.net/index.php/Social_Network_Generation`), we collected examples of real social networks from Pajek, UCINET, Complex Networks, Graph Drawing contests and the InfoVis 2004 contest. To select a representative subset of networks, we drew a portrait of each dataset's characteristics: vertex number (S), edge number (E), density (D) and clustering coefficient (CC). Controlling the average path length is difficult, and we also did not attempt to control degree distribution, considering that our small networks might not have enough vertices to exhibit a power-law degree distribution. Therefore, we attempted to counterbalance S, CC, and D while including networks representative of all three structures we identified (almost-trees, small-world, and almost-complete). We excluded three kinds of graphs for which we considered the results obvious or already proven in Ghoniem et al. experiment: very small graphs containing less than two dozen vertices, small and very sparse graphs for which NL performs very well, and very dense graphs obviously better visualized with MAT. We selected the following datasets: *Infovis* (S=47, E=114, D=0.23, CC=0.83), *Dolphins* (S=47, E=202, D=0.30, CC=0.42), *Fraternity* (S=47, E=294, D=0.36, CC=0.72), *Genealogy* (S=94, E=192, D=0.15, CC=0.59), *Collaboration* (S=94, E=313, D=0.19, CC=0.90) and *USairports* (S=94, E=990, D=0.33, CC=0.84).

To minimize extraneous interpretation issues, we anonymized all actors. To reduce memorization effects, we assigned different labels when presenting the same graph twice. To match the selected graphs in size, we produced a set of randomly filtered graphs for each instance and selected the filtered graphs with D and CC properties similar to the originals. We relaxed the constraint on D when shrinking large graphs, as it is difficult to keep a low D with a high CC when filtering it.

4.3 Experimental setup

Adding interactive highlighting can affect the performance of visualizations. For example, finding the shortest paths between two actors can become preattentive with highlighting. For these reasons, we decided to augment both MAT and NL with interactive highlights to run a fair comparison with MatLink. Subjects were limited to three interactions: mousing over an element, clicking on an element, and dragging on several elements using the left mouse button. Clicking on a row or column header selected one element, indicated by displaying it in red. Similarly, dragging the mouse over a group of elements selected them all. Only vertices were clickable, not edges. In addition, the space bar was used to start and terminate each trial.

We drew NL diagrams using the LinLog algorithm [12]. To reorder matrices, we chose the augmented TSP algorithm described in [10]. We were careful to maintain readability of vertex labels in all conditions. We used the same node sizes for small and large graphs. Since NL visualizations are usually more compact than MAT, we considered it fair to enlarge their nodes to increase their readability.

4.4 Subjects and Apparatus

We used a total of thirty-six subjects, divided into two distinct groups of eighteen. The first group was composed of students and researchers from Univ. of Sydney, mostly from a graph drawing research group. The second group consisted of students and researchers from Univ. of Paris-Sud, primarily specialists in HCI. Only one subject in each group was familiar with matrix-based representations. Both groups used a 3GHz Pentium IV computer with 1GB of RAM and one 19" screen oriented vertically. The screen was divided in two parts: the upper part displayed the question and the lower the visualization.

4.5 Procedure

Before starting the experiment, subjects were given a brief interview to gather information about their previous experience with graphs and visual representations. A tutorial sheet introduced the visualizations, and an experimenter demonstrated the experimental environment, how each representation worked and how to complete the tasks. Subjects could then practice with the program for a few minutes. The training ended when the subject answered all questions correctly for all representations. Subjects spent an average of forty minutes on the training, usually longer than they did on the experiment itself.

The rule for the experiment was to answer correctly as rapidly as possible. If the subjects felt unable to answer a question, they were allowed to skip it. Each subject had a total of 90 questions to answer: 5 tasks performed on 3 visualizations of 6 graphs. To limit the experiment duration, the system limited the completion time (*Time*) to 60 seconds. To limit the subject fatigue, we split the experiment in two sessions with a ten-minute break in between. Moreover, subjects could rest after any question by delaying the start of the next trial.

Sessions were not balanced: the first session presented small datasets, while the second presented large datasets. Sessions were split into three blocks, one for each representation. The order of the representations was complete and counterbalanced across subjects. Each representation block was split into three blocks of three datasets (small for the first session, large for the second) counterbalanced across subjects using a Latin square. We alternated the order of representations to reduce memorization effects: subjects remembering the answer from the previous representation and dataset. However, we kept the order of datasets constant for each session and counterbalanced across subjects.

At the end of the experiment, subjects had to answer a questionnaire about their use of each representation, rating them by task and preference. Finally, a debriefing was conducted to answer any remaining questions and collect their final comments.

4.6 Data collected

Our primary measure was answer correctness. We scored answer correctness on a scale of 0-3, with 0 meaning error and 3 the best answer. For example, analysts exploring a large network are interested in finding primary actors, but while finding the most important rated a 3, other main actors got partial credit. We consider *Time* as a secondary measure. In our analysis, it is reported for all trials, even those with errors. Therefore, caution should be used in evaluating this indicator where the error rate is high, because a fast wrong answer is not useful for data analysis.

5 Results

We performed an analysis of variance (ANOVA) whose results are reported in Table 1. We report Tukey's HSD Post Hoc test (TT) on all significant effects. Table 2 shows mean *Score*, *Time* and Errors.

5.1 All Tasks

Score and Time: A four-way ANOVA *Vis*Task*S*D* reveals a significant effect for *Vis* on *Score* ($F_{40,1400} = 29.7944, p < .0001$) and on *Time* ($F_{40,1400} = 235.5474, p < .0001$). Figure 2a shows the average *Score* per visualization, and TT reveals that MatLink performs significantly better than MAT and NL. Figure 2b shows the average *Time* per visualization. Clearly the speed of NL is much higher than both MatLink and MAT, although TT also indicates significance for the advantage of MatLink over MAT. Figure 2c shows the average number of errors per Task and *Vis*. This information is important to avoid misleading interpretations of the *Time* for Tasks and Visualizations with a high number of Errors. As predicted, MatLink reduces the number of errors produced with MAT for the connectivity tasks (T1 and T2). MatLink produces less errors on all Tasks except T4 (articulationPoint).
User Feedback: Overall, for both group and all tasks, 18/36 users preferred MatLink, 13/36 preferred NL and 4/36 could not decide between MatLink and NL. All users spontaneously reported that they liked MatLink at first sight. After the experiment, they commented that they preferred NL for sparse networks and MatLink for dense networks. Most commented that MatLink was a good compromise for all tasks and graphs except for the articulation point (Task 4). We noticed a difference between groups. Subjects with a graph drawing background showed equal interest in NL and MatLink (8/18 MatLink, 8/18 NL) whereas those from the HCI field preferred MatLink (11/18 MatLink, 5/18 NL).

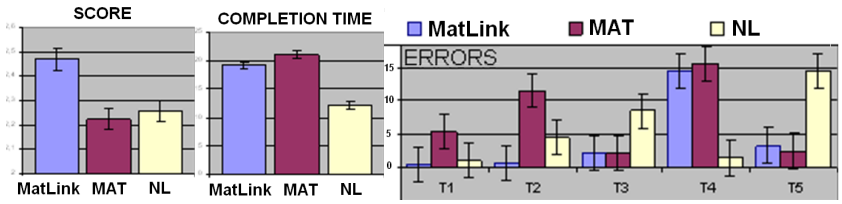
5.2 Task 1 and 2: Connectivity tasks

Predictions: We predicted that for both connectivity tasks (commonNeighbor and shortestPath) MatLink *Score* would be better than MAT, and that it would not be affected by *S* and *D*. We also expected that MatLink would be faster than NL.

		Task 1 commonNeighbor	Task 2 shortestPath	Task 3 mostConnected	Task 4 articulationPoint	Task5 largestClique
Score						
Vis	F(2,70)	11.7262 ***	48.2368 ***	70.2711 ***	068.2356 ***	36.9626 ***
S	F(1,35)	1.4318	18.2947 ***	9.5485 **	0.9167	1.1485
D	F(1,35)	0.5154	2.8872	71.2042 ***	147.0766 ***	74.4772 ***
*S*D*	F(1,35)	8.2470 *	0.7724	0.5085	18.5628 ***	1.4031
Vis**S*	F(2,70)	0.6157	11.8654 ***	8.6931 ***	1.2031	1.275
Vis**D*	F(2,70)	0.902	8.7654 ***	19.032 ***	30.9444 ***	9.2798 ***
Vis**S*D*	F(2,70)	7.7745 ***	0.0335	5.5856 **	2.2344	17.9178 ***
Time						
Vis	F(2,70)	83.4673 ***	56.4941 ***	17.0372 ***	206.8297 ***	5.9018 **
S	F(1,35)	11.2244 ***	218.5899 ***	14.8107 ***	14.7163 ***	30.9148 ***
D	F(1,35)	31.0351 ***	93.5538 ***	36.2844 ***	248.3838 ***	152.5933 ***
*S*D*	F(1,35)	6.2936 *	74.9258 ***	0.0009	40.4718 ***	1.3117
Vis**S*	F(2,70)	1.0601	8.0762 ***	0.8878	4.4137 *	3.3366 *
Vis**D*	F(2,70)	0.207	0.963	5.2016 **	49.5183 ***	0.3993
Vis**S*D*	F(2,70)	8.7874 **	11.0777 ***	1.9474	0.3012	8.4249 ***

$^{***}p < .0001$ $^{**}p < .01$ $^{*}p < .05$

Table 1: ANOVA for each Task for *Vis*, *S*(Size), *D*(Density) and their interactions.



(a) (b) (c)

Fig. 2: Overall average *Score* and *Time* by visualization. Average number of error by task and visualization.

	Task 1 commonNeighbor	Task 2 shortestPath	Task 3 mostConnected	Task 4 articulationPoint	Task5 largestClique	All
Score						
MatLink	**2.95** (0.05)	**2.94 (0.08)**	**2.72 (0.07)**	1.84 (0.09)	**1.89 (0.08)**	**2.47 (0.04)**
Mat	2.64 (0.05)	1.99 (0.08)	**2.68 (0.07)**	1.81 (0.09)	**2.00 (0.08)**	2.19 (0.04)
NL	**2.92 (0.05)**	2.52 (0.08)	1.72 (0.07)	**2.91 (0.09)**	1.20 (0.08)	2.24 (0.04)
Time						
MatLink	10.08 (0.66)	**15.97 (0.78)**	15.15 (0.80)	32.66 (1.32)	22.18 (1.25)	19.12 (0.58)
Mat	17.04 (0.66)	22.70 (0.78)	**11.42 (0.80)**	33.74 (1.32)	20.80 (1.25)	21.44 (0.58)
NL	**7.91 (0.66)**	**15.22 (0.78)**	**11.06 (0.80)**	**8.55** (1.32)	18.43 (1.25)	**12.24 (0.58)**

Table 2: *Score* and *Time* for all visualizations and tasks. Average (Standard Deviation)

Score: ANOVA reveals a significant difference by *Vis*. As expected, TT shows that MatLink is significantly better than MAT for both connectivity tasks. MatLink and NL are not significantly different for commonNeighbor, but surprisingly MatLink performs significantly better than NL for shortestPath. *D* has no significant impact for either task and visualizations. *S* has a significant effect on shortestPath: subjects have higher scores on small graphs for all visualizations. ANOVA reveals that interactions *Vis*S* and *Vis*D* are significant for shortestPath; MAT is particularly affected by both of them. *Vis*D* reveals a surprising effect for NL, where users had higher scores for higher *D*.
Time: ANOVA reveals a significant effect on *Vis*, *S* and *D*. As expected, TT shows that MatLink is significantly faster than MAT for both tasks. However, MatLink is only significantly faster than NL for the shortestPath. We predicted that the *Time* would increase for large and dense graphs on both tasks. However, the interaction *Vis*S* for shortestPath shows that all visualizations including MatLink are affected by *S*.
User Feedback: Most of the users preferred MatLink.

5.3 Task 3: Most Connected Actor

Prediction: We expected MatLink to perform as MAT both in term of *Score* and *Time*. We expected MAT to be faster than NL and not affected by *S* or *D*.
Score: ANOVA reveals significant effects of the three independent variables *Vis*, *S* and *D*. TT shows that MatLink and MAT produce better scores than NL. This task is affected both by *S* and *D*: the large and dense the graphs are, the lower the *Score* for all visualizations. Both interactions *Vis*S* and *Vis*D* reveal that NL is especially affected by these variables.
Time: This indicator is to be interpreted carefully for NL as the error rate is 28% for this task. ANOVA reveals significant effects for *Vis*, *S* and *D*. For all visualizations, *Time* increases with *S* and with *D*. TT shows that MatLink is slower than the other representations. The interaction analysis of the interaction *Vis*D* reveals that MatLink is particularly affected by *D* whereas the interaction *Vis*S* is not significant.
User Feedback: Most of the users preferred MAT. Several subjects report that MatLink lacked a highlight or mouse-over effect to ease the comparison of actors, especially to help them counting the number of edges in the matrix row/column.

5.4 Task 4: Articulation Point

Prediction: We predicted that NL would outperform MatLink and MAT both in term of *Score* and *Time*. We expected MatLink to show higher Scores than MAT especially for sparse graphs.
Score: As predicted, ANOVA reveals significant effect of *Vis*. TT shows that NL produces significantly higher scores. The *Score* for this task is affected by *S* and by *D*. The interaction *Vis*D* reveals that MatLink and MAT are affected by *D* for this task: dense graph produce significantly lower scores for both representations. Unexpectedly, ANOVA reveals no significant difference between MAT and MatLink for sparse or dense graphs.

Time: ANOVA reveals that *Vis*, *S* and *D* have a significant impact on the *Time*. TT confirms that NL outperforms significantly MAT and MatLink for this task.
User Feedback: All subjects except two preferred NL for this task. Their level of confidence was very high and their performance good with this representation. Almost all subjects report that this task was the most difficult one when dealing with MAT and dense graphs. Several subjects commented that MatLink was slightly better for sparse graphs.

5.5 Task 5: Clique

Prediction: We expected MatLink to perform slightly better than MAT in terms of *Score*. We also predicted NL would present low scores for dense graphs. We did not expect any significant difference in *Time*.
Score: ANOVA reveals significant difference by *Vis* and *D*. As expected, TT shows that MatLink and MAT produce higher scores than NL. However, ANOVA does not reveal any significant difference between MatLink and MAT. TT shows significantly smaller scores for all visualizations with dense graphs. The interaction *Vis*D* shows that NL scores are significantly reduced on dense graphs.
Time: ANOVA reveals a significant difference by *Vis*: NL is faster than MatLink and MAT. However, this indicator should be interpreted carefully, because NL's error rate is 48% for this task: users who gave up immediately were counted as having a fast (if erroneous) completion. ANOVA shows also that *S* and *D* affect the *Time*. The interaction *Vis*D* is not significant, whereas the interaction *Vis*S* is significant: MatLink and MAT are slower when *S* increases.
User Feedback: Almost all subjects preferred MAT or MatLink for this task. They comment that they had a much higher level of confidence identifying if a visual cluster was a clique or not. However, several added that finding the largest clique was difficult without reordering the MAT to place sub-parts next to each other. Most reported that links were not useful, but several argued that links helped them find a member of the clique distant from the others in the representation.

6 Discussion

For the social networks and tasks we studied, users were significantly more accurate with MatLink than with MAT for most tasks. Although NL is usually more accurate than MAT for path-based tasks, MatLink was more accurate than NL for one task in our study (shortestPath). MAT was known to be more accurate than NL on dense and large graphs for mostConnected, but on this dataset it was also better for largestClique. Using the best layout for each representation, MAT and MatLink were both better than NL for the clique-finding task. Moreover, we confirmed Ghoniem et al. [1] results that MAT outperforms NL for low-level tasks on our selected social networks.

As we predicted, for the path-related tasks (1 and 2), MatLink performance is much better than MAT and is competitive with NL, outperforming it for the shortest path task even when the path is highlighted. A surprising detail is that

users had fewer errors with NL on dense graphs than sparse ones. This appears to be an artifact of the highlighting of the shortest paths. From our observations, it seems that users ignored the highlighting information in our NL implementation when the network was sparse and thus missed the shortest path, whereas they systematically used highlighting on dense graphs.

For task 3 (mostConnected), as predicted, users were more accurate using MAT and MatLink than with NL. However, surprisingly, the time was longer for MatLink than MAT. Our interpretation is that the added overhead of drawing the auxiliary visualizations created a small time lag that impacted this task. We did not provide continuous feedback about the vertex under the mouse, and several users complained about the lack of this feature when they clicked near but not on the most-connected vertex.

Interestigly, although we expected a difference between the two groups of subjects, one of which had a strong background in graph drawing using NL, our results do not show a significant difference between groups. Users usually performed tasks faster with NL than with MAT and MatLink, but this finding should be interpreted carefully, because users had many errors with NL than with MatLink and MAT (except for task 4 articulationPoint).

7 Conclusions

This article describes and evaluates MatLink: an enhanced matrix visualization of graphs that overlays a linear node-link representation on the matrix and adds dynamic feedback of path-relationship between nodes.

We assessed the effectiveness of this visualization by a controlled experiment comparing user performance for representative social network analysis tasks, performed on realistic social networks fitting in full screen. The three tools assessed were MatLink, conventional matrix representations (MAT), and node-link diagrams(NL). The experiment showed that MatLink significantly outperforms ordinary matrix visualizations for path-related tasks — known to be difficult on MAT — that are required for social network analysis. It also performs as well or better than MAT on other analysis tasks that cannot be effectively performed on NL because of the inherent tendency of social networks to be locally dense. Therefore, MatLink is a good compromise for visualizing and analyzing social networks with a single representation. However, on some tasks related to overall graph structure, NL is still superior.

Other improvements are possible for all three visualizations. For example, we assigned shortest-path for the dynamic feedback, but other kind of paths could be computed and highlighted based on user preferences. We also need to assess the effectiveness of MatLink for larger networks when scrolling or other navigation techniques are required. Now that users can visualize and analyze denser social networks, we need to understand better the kind of supports they will need. Longitudinal user studies would help us understand user needs and preferences. Which representation will they choose when all are available? For which tasks? Will they alternate between representations? How often?

8 Acknowledgments

We would like to thank Howard Goodell for improvements to the article and to Yann Riche for helping with the accompanying video.

References

[1] Ghoniem, M., Fekete, J.D., Castagliola, P.: On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. Information Visualization **4**(2) (2005) 114–135
[2] Wasserman, S., Faust, K.: Social Network Analysis. Cambridge Univ. Press (1994)
[3] Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature **393** (1998) 440 – 442
[4] Amar, R., Eagan, J., Stasko, J.: Low-level components of analytic activity in information visualization. In: Proceedings of the IEEE Symposium on Information Visualization. (2005) 111– 117
[5] Plaisant, C., Lee, B., Parr, C.S., Fekete, J.D., Henry, N.: Task taxonomy for graph visualization. In: BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV'06), Venice, Italy, ACM Press (2006) 82–86
[6] Heer, J., Boyd, D.: Vizster: Visualizing Online Social Networks. In: Proceedings of the IEEE Symposium on Information Visualization. (2005) 5
[7] de Nooy, W., Mrvar, A., Batagelj, V.: Exploratory Social Network Analysis with Pajek. Structural Analysis in the Social Sciences. Cambridge Univ. Press (2005)
[8] Adar, E.: Guess: a language and interface for graph exploration. In: CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems, New York, NY, USA, ACM Press (2006) 791–800
[9] Doreian, P., Batagelj, V., Ferligoj, A.: Generalized Blockmodeling. Structural Analysis in the Social Sciences. Cambridge Univ. Press (2005)
[10] Henry, N., Fekete, J.D.: MatrixExplorer: a Dual-Representation System to Explore Social Networks. IEEE Transactions on Visualization and Computer Graphics **12**(5) (2006) 677–684
[11] Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for the Visualization of Graphs. Prentice Hall PTR (1998)
[12] Noack, A.: Energy-based clustering of graphs with nonuniform degrees. In Healy, P., Nikolov, N.S., eds.: Proceedings of the 13th International Symposium on Graph Drawing (GD 2005), Limerick, Ireland, Springer-Verlag (2005) 309–320
[13] Auber, D., Chiricota, Y., Jourdan, F., Melancon, G.: Multiscale visualization of small world networks. In: Proceedings of the IEEE Symposium on Information Visualization. (2003) 75–81
[14] Wattenberg, M.: Visual exploration of multivariate graphs. In: Proceedings of the SIGCHI conference on Human Factors in computing systems, Montréal, Québec, Canada, ACM Press (2006) 811–819
[15] Bertin, J.: Semiology of graphics. Univ. of Wisconsin Press (1983)
[16] Abello, J., van Ham, F.: Matrix zoom: A visual interface to semi-external graphs. In: Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'04), Austin, Texas (2004) 183–190
[17] van Ham, F.: Using multilevel call matrices in large software projects. In: Proceedings of the IEEE Symposium on Information Visualization, Seattle, WA, USA (2003) 227–232
[18] Fekete, J.D.: The InfoVis Toolkit. In: Proceedings of the IEEE Symposium on Information Visualization. (2004) 167–174