Matrix Snap&Go: Visualization of Paths on Matrices

Z. Huang[1], D. Archambault[2], R. Borgo[3], and A. Kerren[1,4]

[1]Linköping University, Sweden [2]Newcastle University, United Kingdom [3]King's College London, United Kingdom [4]Linnaeus University, Sweden

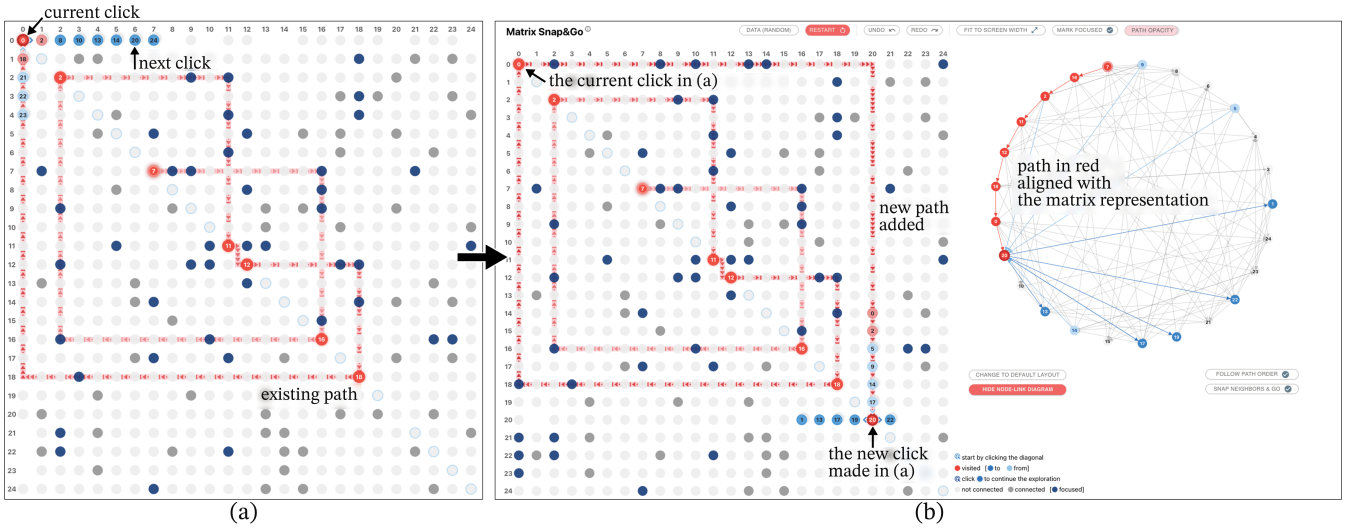

Figure 1: *Exploring the matrix representation of a directed graph. Direct neighbors of the currently selected node are drawn close to the node, with the user's past interaction history highlighted in red. Screenshot (a) shows an initial path established by previous user interactions; whereas (b) provides an overview of our Matrix Snap&Go approach, resulting from (a) by clicking one of the adjacent nodes ("next click"). In (b), left: the adjacency matrix visualization; right: a radial node-link diagram visualizing the same input graph and synchronizing the same interaction process. The node-link diagram is present on the right only to clarify how to read a path on a matrix using our approach.*

Abstract

Matrix representations can be effective for visualizing networks. However, it is very difficult to follow or explore specific paths in a matrix representation. In this paper, we introduce an interactive method for exploring paths on a matrix, called Matrix Snap&Go. Our visualization approach relies heavily on interactive exploration, bringing in the local neighborhood of selected nodes and tracing the path progression through the matrix. We demonstrate the utility of our approach by performing and analyzing test runs with synthetic input graphs of various node/edge densities as well as by discussing a use case based on the exploration of citation networks.

CCS Concepts

• ***Human-centered computing*** → *Graph drawings; Information visualization;*

1. Introduction

Matrix visualizations of networks/graphs (we use both terms synonymously in this paper) support many graph reading tasks and offer advantages over standard node-link diagrams in various scenarios. For instance, adjacency matrix representations can address issues of occlusion and edge clutter common in node-link layouts, by representing connections through cells at the intersection of a row (source node) and a column (target node). Consequently, using matrices to read graphs has become widely used across multiple research domains [BBHR*16, MGM*19]. In social network analysis, for example, viewing graph data in matrix format has been a common practice for gaining insights into meso-level statistics of actors or identifying structural patterns like cliques [HF06, HF07].

However, among many other challenges, matrix representations of graphs face several common issues. For instance, compared to node-link layouts, they are often seen as less flexible [BBDW14].

DOI: 10.2312/evs.20241058

Previous research on matrix interaction and animation has focused on block modeling, with relatively less attention given to other interactive elements in the design space. Moreover, reducing cognitive complexity during matrix exploration—such as locating edges and preserving the mental map during layout animations—remains challenging. This leads to an important limitation of a standard matrix visualization, which we address in this paper, which is reading paths along the graph, as only direct adjacencies are represented by an adjacency matrix. One way to address this issue is to augment the visualization of paths within a matrix visualization through interaction, allowing users to explore and visualize the structure of a path interactively on top of the matrix.

To address the problem of visualizing paths on matrices, we propose *Matrix Snap&Go*: an interactive technique to help users create paths on matrices and visualize the history of that path through a matrix representation. *Matrix Snap&Go* makes effective use of spatial position, bringing in the local neighborhood of a node in order to understand future exploration directions for the path through the matrix visualization. The history of the path is explored through a polyline that is able to demonstrate both the nodes and edges that have been traversed in order to achieve that path. Our contributions in this paper are as follows:

- a novel interactive, matrix path visualization technique.
- a user-guided reordering interactions on the adjacency matrix visualization, making the path easier to follow on the matrix.
- a demonstration of the usability of the approach by testing on a number of real and synthetically generated datasets and showing its advantages and limitations. A live demo can be found at `ivis-tools.itn.liu.se/matrix-snap-and-go`.

The remainder of this paper is organized as follows: Section 2 introduces related work in matrix visualization and problem definition. Section 3 describes system design and implementation details. Section 4 presents cases where *Matrix Snap&Go* is tested and being used, as well as discusses limitations and future work. Finally, Section 5 concludes our work.

2. Background

Matrix visualizations have numerous perceptual advantages especially for dense graphs [GFC05]. As such, a number of techniques have been explored for visualizing series of matrices [BPF14] and comparing differences and similarities [BHRD*15, VKA*18]. Performing analytical tasks using these visualizations often requires advanced interactive techniques. Network visualization incorporates multiple levels of interaction, including view-level, visual-structure, and data-level interaction [WEF*14]. Adjacency matrix visualization, in particular, relies heavily on interacting with visual structures, such as cells, columns, and rows [PDF14]. There are two main objectives for such interactions: navigating and showing network details or producing block-diagonal patterns. Users can interact with a matrix visualization by manually reordering rows and columns to change the visual style directly, or by steering existing reordering algorithms to identify, select, and adjust matrix substructures [BSP20]. However, if the underlying data lacks such inherent patterns, many interaction algorithms may struggle to reveal any significant structure [BTBC*21, ASA*22]. As a result, there is an ongoing challenge in designing alternative interaction techniques that focus on defining and discovering specific statistical, global, or local patterns.

Pathfinding on matrices Many studies have shown that matrices are ill-suited for pathfinding and its related tasks [GFC05, MGM*19]. Unlike node-link diagrams, where the spatial arrangement between nodes can convey additional information, the visibility of paths in a standard matrix depends heavily on the quality of its ordering. The majority of existing work has approached this problem by finding the optimal ordering or combined matrices with node-link diagrams to reveal structural features, such as the shortest paths [HF07,HFM07,VBW15]. Different from uncovering inherent path-related structural properties in a graph, we present an interaction technique that allows users to actively or passively create paths during their exploration of the matrix and visualize the results.

Our title and technique are inspired by Bring&Go, which is a node-link diagram navigation technique for paths where the local neighborhoods of networks are brought in around the current node in the exploration [MCH*09]. Here, the user selects a destination node, and an animation smoothly transitions between the current and destination nodes demonstrating how the path is constructed. One could view our approach as a similar technique for path exploration, but for matrix visualizations of networks. Other related approaches focus on the exploration of multiple networks in different views such as Hub2Go [ZSK17]. Probably the closest paper to our own is that of Shen and Ma [SM07]. In their work, paths are visualized on top of adjacency matrices through lines. A path is considered a series of edges, and cells of the matrix—the edges of the graph—are connected through horizontal and vertical lines. Thus, multiple paths can be visualized using this technique. In our work, we focus on user-constructed paths through the adjacency matrix and the interaction required when creating such a path. Also, in contrast to the work [SM07], both nodes and edges are visualized through the *use of the diagonal* of the matrix representation.

3. Implementation

We have designed and implemented *Matrix Snap&Go*, a novel visualization approach for user-centric, matrix-based graph exploration. It provides visual cues and records directional paths of visited nodes and edges (Figure 1). This section discusses the visual encoding and interaction details behind the *Matrix Snap&Go* interface, which takes a generic graph as input but excludes self-loops.

The system interface (Figure 1b) consists of two main components: an action bar with customization options at the top and a matrix view on the left showing the core interaction features of *Matrix Snap&Go* which is our technique. A radial node-link view on the right provides training on how to read our technique. The layout is optimized for directed graphs while also accommodating undirected graphs. Both the left and right panels represent the same input graph. They are synchronized to display a user's present and past selections and their exploration path. The proposed adjacency matrix visualization aligns with how people typically interpret such representations (Figure 1a) [GFC05]. If an edge exists from node a to node b, the cell at the intersection of row a and column b will be marked in a ● dark grey color to show the connectivity compared to ● unconnected cells. Since the ● diagonal of the matrix

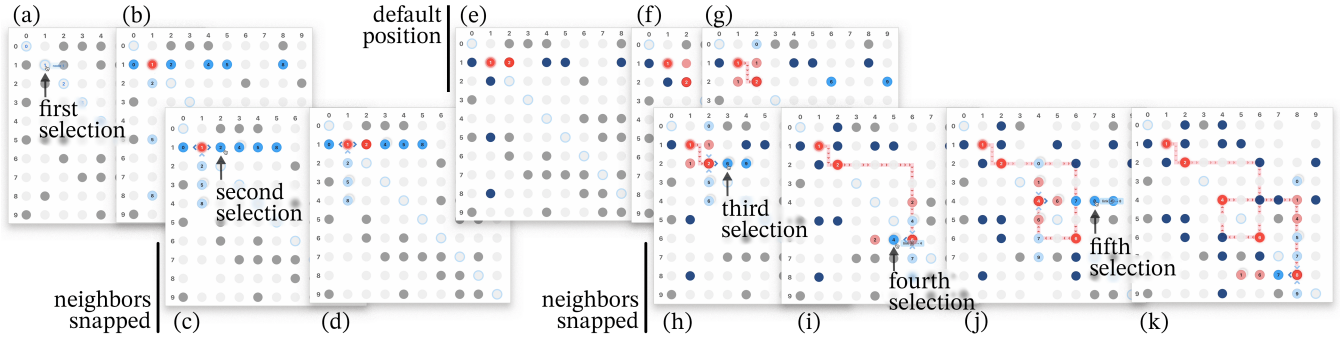

Figure 2: *Eleven matrices arranged in alphabetical order from (a) to (k). They capture the evolution of the matrix layout through a sequence of five user interactions. The figures are arranged to highlight changes with minimal space, resulting in a compact, irregularly overlapping horizontal layout. Vertically, the top subfigures depict the matrix in its default state, using colors to mark node statuses; while the bottom subfigures show reordering phases, with selected neighbors pulled closer (a larger version is available as supplementary material).*

represents nodes connecting to themselves, we treat these cells as individual nodes rather than self-loops.

Matrix Snap&Go maintains consistent color encodings across both the matrix and node-link diagram representation of the underlying graph. We use the color red to denote the path between two selected nodes, as shown in Figure 1. Selected nodes are marked on the matrix diagonal with a icon, with the first node in the interaction sequence more distinguishable with . The non-diagonal cells of the matrix represent links between nodes. Upon selecting a node, its outgoing links are highlighted with darker blue , while incoming links are indicated with a lighter blue color . User interaction is further constrained, as one can click on outgoing links to navigate forward, but in most cases cannot move backward against the link direction to ensure a cohesive user experience that aligns with the link directionality in a directed graph. Additionally, a lighter red color indicates when a link's target or source node has been previously selected. We also introduce to mark all links connected to nodes that were "snapped" into focus, which will be explained next.

3.1. *Matrix Snap&Go* Algorithm

Initialization Users choose a starting node to initiate the exploration process by selecting a diagonal cell in the graph's adjacency matrix (Figure 2a). The selection is restricted to diagonal cells to ensure a single exploration path is established.

Algorithm phases Figure 2(c–h) presents a decomposed version of our proposed interaction technique. *Matrix Snap&Go* consists of three straightforward steps, with animations integrated. **1. Snap:** When a user selects a node, the links connected to this node are highlighted and moved next to the current selection (Figure 2(b–c)). The term "snapped" refers to this highlighting and reordering process. **2. Go:** Clicking an outgoing link (non-diagonal cells) shifts the focus from the initial node to the target node of the clicked link, while still acknowledging the initial selection. The focus moves to the target node, which is now considered the current selection. During this phase, as shown in Figure 2(d–f), the currently selected cell reverts to its original, default diagonal position, indicating the target node it directs to. Concurrently, in (e), the adjacent nodes previously "snapped" revert to their starting positions as shown in Figure 2a. **3. Pathfinding:** This step creates a visual path connecting the previously selected node with the current selected node, and also passing through (or representing) the selected edge in between (Figure 2(f–g)). The path direction is indicated by red chevrons, which guide the selection from one node to the next, corresponding to the sequence of those selections (Figure 2(h–k)).

In the following, we provide a more detailed discussion of our design decisions taken with respect to these three phases.

Layout simulation of the neighborhood After the initial selection, edges connected to the selected node are highlighted in blue , indicating their connection to the focus node (Figure 2b). Subsequently, these highlighted edges are moved closer to the focus node as shown in Figure 2c). To ensure consistency, the neighbors are ordered according to their original sequence in the columns/rows (as in Figure 2a). One advantage of this specific representation for a directed graph is that, it enhances readability to differentiate between incoming links, positioned vertically to the focus node , and outgoing links, positioned horizontally . Especially in sparse matrices without other pre-defined reordering algorithms, pulling neighbors closer reduces the cognitive load in identifying paths and finding neighboring nodes for creating potential paths. Additionally, when the matrix becomes larger and occupies a large portion of the screen space, identifying what each cell represents becomes more challenging. Thus, this rearrangement process enforces the user's attention on local neighborhoods.

When edges are shifted, their original positions may become empty (for example, the bottom part of the second column in Figure 2c). However, it is often the case that the target position next to the focus node is already occupied by an unhighlighted cell, indicating no direct connection between the focus node and the node at the other end of the link the cell represents . In such cases, the unconnected cell is adjusted slightly upwards and rightwards. This creates an offset, making it seem as though the cells are layered, resembling two cards stacked on top of one another .

Layout transformation for a new edge selection To illustrate the transition from the first node to the second via the selected edge, we first animate resetting the previously highlighted neighboring links of the first node (Figure 2(d–e)), then animate the transition of

the selected edge to the newly focused node (Figure 2(e–g)). Since diagonal positions represent nodes, the cell selection changes its representation meanings from links to nodes so it shifts to the diagonal. Then the neighbors of the newly selected node are snapped, and the algorithm iterates upon each user selection (Figure 2(c–h)). Additionally, a different color ● is used to mark all links that have been snapped closer to highlight that they have received more attention from the user than other connected cells.

Path drawing By visualizing a thread touching all visited edges and nodes through the diagonal, the exploration history of a user can be recorded. In most cases, we draw straight lines between two cells based on the Manhattan distance with minimal turns. However, a trade-off exists between identifiable paths that minimize overlap and the precise alignment with the selected edge, so additional considerations for rerouting are introduced. If two cells are diagonally neighbors, an "s" shape path is drawn instead of a Manhattan path with one turn . Otherwise, there is a complete overlap of the path with itself. In a matrix visualization, cells symmetrical to the diagonal indicate connections between the same pair of nodes with opposite directionality. By default, paths are drawn in the upper triangular matrix . However, the choice of drawing either in the upper or lower triangular depends on which path minimizes the overlapping pixels of the existing paths (Figure 2).

Node-link diagram Alongside the matrix, users can choose to optionally view a force-directed or radial node-link diagram to help them learn the snap&go concept applied to alternative graph representations (Figure 1b). With each node selection, the radial node-link diagram animates, pulling neighboring nodes closer to the chosen node then returning them to their original positions. Afterward, the newly selected node is positioned adjacent to the previously selected one, making the selection sequence path distinct to read.

4. Usage Scenarios and Discussion

Matrix Snap&Go can be applied for various use cases besides synthetic graphs, including integrating into an existing visual analytics pipeline. Figure 3 showcases the system for tracking the exploration history of papers based on their citation network (data from [IHK*17]). It is currently set to a zoomed-in view of the matrix visualization with the browser window auto-locating to every new selection, and users can zoom out to obtain an overview of the existing path at any time. We applied the k-core algorithm [BZ03] to extract a subset of IEEE vis papers that frequently cite one another. The system displays information like paper titles and DOIs when the mouse hovers over the cells. *Matrix Snap&Go* captures the process of finding a paper to read, reading one of its citations, and then recording the snowballing process as an exploration path on a matrix. We enabled clicks on both incoming and outgoing links of a node to accommodate users' interest in exploring both cited and citing papers. This click restriction can be customized as needed. One might also use filtering or subgraph extraction techniques to get subsets for specific tasks or combine other matrix ordering techniques [BBHR*16] to customize the layout.

Limitations and future work We reviewed the proposed tool, *Matrix Snap&Go*, using various directed graphs generated by the Erdős-Rényi model [ER*60]. Figure 1 illustrates the system layout given a directed Erdős-Rényi graph with 25 nodes and 0.2 probability for edge creation. Through trying different numbers of nodes and graph densities, we found that the system performs best with a graph density between 0.05 and 0.5. It becomes relatively less effective for dense graphs, because the visibility of snapped cells decreases when many adjacent nodes are connected. Previous studies have indicated that matrices are not efficient for representing sparse graphs [OJK18], but for this case our system provides interactive methods to highlight neighboring nodes adequately. *Matrix Snap&Go* is designed to simplify the immediate exploration of connections by displaying all available next steps within a matrix layout. And it enables the possibility of introducing new panels in future work to show additional contextual information to guide users in choosing a path.

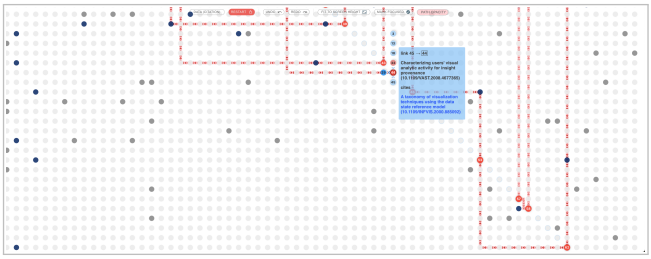

Figure 3: *Matrix Snap&Go applying zoomed-in layout with auto-relocation, and visualizing a citation network of 70 nodes.*

Our current visualization prototype is web-based and designed to handle graphs with 10 to 100 nodes to ensure timely responses. Although it can manage more nodes, response times may increase depending on the browser's capacity. Future work to enhance scalability could involve summarizing the matrix cells into blocks, reordering a block of nodes/links rather than a single link, and providing details on demand to zoom in on specific communities. In addition to scalability, the evaluation of *Matrix Snap&Go* is another direction of future work. While we presented a use case of citation networks where favoring matrix representations over node-link diagrams provides a better reading of the directionality of a chosen ego network, it would be valuable to conduct formal user studies across various usage scenarios to compare and evaluate the readability and efficiency of this *Snap&Go* interaction with different visual representations over different data distributions.

5. Conclusion

In this paper, we presented *Matrix Snap&Go*, a visualization of exploration paths on adjacency matrices. Our animated interaction technique supports users in exploring matrices by offering proper guidance, which provides a good starting point for future work. We have also demonstrated the usefulness of *Matrix Snap&Go* and discussed its advantages and limitations.

Acknowledgements The authors wish to thank Kostiantyn Kucher for his detailed and valuable comments on the visualization design and paper structure. This work was partially supported through the ELLIIT environment for strategic research in Sweden. We thank Dagstuhl Seminar #23051 "Perception in Network Visualization" for initial discussions on this paper [KKR*23].

References

[ASA*22] ABDELAAL M., SCHIELE N. D., ANGERBAUER K., KURZHALS K., SEDLMAIR M., WEISKOPF D.: Comparative evaluation of bipartite, node-link, and matrix-based network representations. *IEEE Transactions on Visualization and Computer Graphics 29*, 1 (2022), 896–906. doi:10.1109/TVCG.2022.3209427. 2

[BBDW14] BECK F., BURCH M., DIEHL S., WEISKOPF D.: The state of the art in visualizing dynamic graphs. In *EuroVis - STARs* (2014), The Eurographics Association. doi:10.2312/eurovisstar.20141174. 1

[BBHR*16] BEHRISCH M., BACH B., HENRY RICHE N., SCHRECK T., FEKETE J.-D.: Matrix reordering methods for table and network visualization. *Computer Graphics Forum 35*, 3 (2016), 693–716. doi:10.1111/cgf.12935. 1, 4

[BHRD*15] BACH B., HENRY-RICHE N., DWYER T., MADHYASTHA T., FEKETE J.-D., GRABOWSKI T.: Small multipiles: Piling time to explore temporal patterns in dynamic networks. *Computer Graphics Forum 34*, 3 (2015), 31–40. doi:10.1111/cgf.12615. 2

[BPF14] BACH B., PIETRIGA E., FEKETE J.-D.: Visualizing dynamic networks with matrix cubes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2014), CHI '14, Association for Computing Machinery, p. 877–886. doi:10.1145/2556288.2557010. 2

[BSP20] BEHRISCH M., SCHRECK T., PFISTER H.: GUIRO: User-guided matrix reordering. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2020), 184–194. doi:10.1109/TVCG.2019.2934300. 2

[BTBC*21] BURCH M., TEN BRINKE K. B., CASTELLA A., PETERS G. K. S., SHTERIYANOV V., VLASVINKEL R.: Dynamic graph exploration by interactively linked node-link diagrams and matrix visualizations. *Visual Computing for Industry, Biomedicine, and Art 4*, 1 (2021), 1–14. doi:10.1186/s42492-021-00088-8. 2

[BZ03] BATAGELJ V., ZAVERSNIK M.: An o(m) algorithm for cores decomposition of networks. *arXiv:cs.DS/0310049* (2003). 4

[ER*60] ERDŐS P., RÉNYI A., ET AL.: On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences 5*, 1 (1960), 17–60. 4

[GFC05] 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. doi:10.1057/palgrave.ivs.9500092. 2

[HF06] 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. doi:10.1109/TVCG.2006.160. 1

[HF07] HENRY N., FEKETE J.-D.: MatLink: Enhanced matrix visualization for analyzing social networks. In *Human-Computer Interaction – INTERACT 2007* (2007), Springer Berlin Heidelberg, pp. 288–302. doi:10.1007/978-3-540-74800-7_24. 1, 2

[HFM07] HENRY N., FEKETE J.-D., MCGUFFIN M. J.: NodeTrix: a hybrid visualization of social networks. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1302–1309. doi:10.1109/TVCG.2007.70582. 2

[IHK*17] ISENBERG P., HEIMERL F., KOCH S., ISENBERG T., XU P., STOLPER C., SEDLMAIR M., CHEN J., MÖLLER T., STASKO J.: vispubdata.org: A metadata collection about IEEE visualization (VIS) publications. *IEEE Transactions on Visualization and Computer Graphics 23*, 9 (2017), 2199–2206. doi:10.1109/TVCG.2016.2615308. 4

[KKR*23] KLEIN K., KOBOUROV S., ROGOWITZ B. E., SZAFIR D., MILLER J.: Perception in Network Visualization (Dagstuhl Seminar 23051). *Dagstuhl Reports 13*, 1 (2023), 216–244. doi:10.4230/DagRep.13.1.216. 4

[MCH*09] MOSCOVICH T., CHEVALIER F., HENRY N., PIETRIGA E., FEKETE J.-D.: Topology-aware navigation in large networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2009), CHI '09, Association for Computing Machinery, p. 2319–2328. doi:10.1145/1518701.1519056. 2

[MGM*19] MCGEE F., GHONIEM M., MELANÇON G., OTJACQUES B., PINAUD B.: The state of the art in multilayer network visualization. *Computer Graphics Forum 38*, 6 (2019), 125–149. doi:https://doi.org/10.1111/cgf.13610. 1, 2

[OJK18] OKOE M., JIANU R., KOBOUROV S.: Revisited experimental comparison of node-link and matrix representations. In *Graph Drawing and Network Visualization* (2018), Springer International Publishing, pp. 287–302. doi:10.1007/978-3-319-73915-1_23. 4

[PDF14] PERIN C., DRAGICEVIC P., FEKETE J.-D.: Revisiting Bertin matrices: New interactions for crafting tabular visualizations. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 2082–2091. doi:10.1109/TVCG.2014.2346279. 2

[SM07] SHEN Z., MA K.-L.: Path visualization for adjacency matrices. In *Eurographics/ IEEE-VGTC Symposium on Visualization* (2007), The Eurographics Association. doi:10.2312/VisSym/EuroVis07/083-090. 2

[VBW15] VEHLOW C., BECK F., WEISKOPF D.: The State of the Art in Visualizing Group Structures in Graphs. In *Eurographics Conference on Visualization (EuroVis) - STARs* (2015), The Eurographics Association. doi:10.2312/eurovisstar.20151110. 2

[VKA*18] VOGOGIAS A., KENNEDY J., ARCHAMBAULT D., BACH B., SMITH V. A., CURRANT H.: Bayespiles: Visualisation support for bayesian network structure learning. *ACM Transactions on Intelligent Systems and Technology 10*, 1 (2018). doi:10.1145/3230623. 2

[WEF*14] WYBROW M., ELMQVIST N., FEKETE J.-D., VON LANDESBERGER T., VAN WIJK J. J., ZIMMER B.: Interaction in the visualization of multivariate networks. In *Multivariate Network Visualization: Dagstuhl Seminar #13201, Dagstuhl Castle, Germany, May 12-17, 2013, Revised Discussions* (2014), Kerren A., Purchase H. C., Ward M. O., (Eds.), Springer International Publishing, pp. 97–125. doi:10.1007/978-3-319-06793-3_6. 2

[ZSK17] ZIMMER B., SAHLGREN M., KERREN A.: Visual analysis of relationships between heterogeneous networks and texts: An application on the IEEE VIS publication dataset. *Informatics 4*, 2 (2017). doi:10.3390/informatics4020011. 2