MULTIAGR: A Technique for Aggregating Multivariate Networks

Pranav Rajan
University of Utah

UUCS-21-011

School of Computing
University of Utah
Salt Lake City, UT 84112 USA

1 May 2021

Abstract

Data analysts and researchers in domains ranging from biology to electrical engineering are using multivariate networks (MVNs) to study the relationships and topology between individual data points and their associated attributes to develop new insights about the relationships and properties in their network data. To visually represent MVNs, two methods are commonly used: 1) Node-Link diagram representation 2) Adjacency Matrix representation. In order to visualize MVN data using these methods, data analysts and researchers perform a series of network wrangling operations to reshape their data to visually analyze a particular question of interest. This process requires familiarity with data wrangling tools such as Python, R, or Excel which presents a barrier to those unfamiliar with these tools and the operations needed to visualize particular features of an MVN dataset. Currently, there are limited available tools that enable users to perform network wrangling operations concurrently with visualizations for multivariate network analysis. In this thesis, we explore a proof of concept technique for the aggregation data wrangling operation on the adjacency matrix representation of an MVN. We develop a new visual aggregation method inspired by PivotGraph [25], called MultiAggr, for aggregating an adjacency matrix representation of an MVN using categorical node attributes.

MULTIAGGR:
A TECHNIQUE FOR AGGREGATING MULTIVARIATE NETWORKS

by

PRANAV RAJAN

A Senior Thesis Submitted to the Faculty of
The University of Utah
In Partial Fulfillment of the Requirements for the

Degree of Bachelor of Science

In

Computer Science

Approved:

Alexander Lex
Thesis Faculty Supervisor

Mary Hall
Director, School of Computing

Jim de St. Germain
Director of Undergraduate Studies

April 2021

ABSTRACT

Data analysts and researchers in domains ranging from biology to electrical engineering are using multivariate networks (MVNs) to study the relationships and topology between individual data points and their associated attributes to develop new insights about the relationships and properties in their network data. To visually represent MVNs, two methods are commonly used: 1) Node-Link diagram representation 2) Adjacency Matrix representation. In order to visualize MVN data using these methods, data analysts and researchers perform a series of network wrangling operations to reshape their data to visually analyze a particular question of interest. This process requires familiarity with data wrangling tools such as Python, R, or Excel which presents a barrier to those unfamiliar with these tools and the operations needed to visualize particular features of an MVN dataset. Currently, there are limited available tools that enable users to perform network wrangling operations concurrently with visualizations for multivariate network analysis. In this thesis, we explore a proof of concept technique for the aggregation data wrangling operation on the adjacency matrix representation of an MVN. We develop a new visual aggregation method inspired by PivotGraph [25], called MultiAggr, for aggregating an adjacency matrix representation of an MVN using categorical node attributes.

For my parents, Sundar and Nandini, and my advisors Alex and Derya.

CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to thank Alexander Lex and Derya Akbaba for advising me on this thesis as well as Jack Wilburn and Roni Choudhury for providing valuable technical feedback and supervising my contributions to MultiNet. I would also like to thank my parents, Sundar and Nandini who supported me during my undergraduate studies, Timothy Kay, who introduced me to visual computing and how to think about problem solving, and the Utah Visualization Design Lab for providing a supportive community and sharing technical knowledge that helped me complete this project.

This work was funded by the University of Utah Office of Undergraduate Research UROP Program (FALL 2021) and the National Science Foundation REU program as part of the NSF funded Collaborative Research: Framework: Software: HDR: Reproducible Visual Analysis of Multivariate Networks with MultiNet project.

CHAPTER 1

INTRODUCTION

As our world has become more connected and information more accessible, networks have become an important tool for modeling and representing real-world phenomena. A network is defined by nodes and links connecting these nodes. Nodes and links are used to represent connected objects such as people, neurons and airports. Examples of real-world networks include social networks, the internet, path analysis and evolutionary species tree networks. Multivariate networks (MVNs) are a special kind of feature-rich network where nodes and links have their own attributes [19]. Attributes can range from a single property such as an airport location for a node in a flight path to attribute vectors containing multiple properties such as total air time, taxi delay time, flight delay time, or layover time for a flight link between two airport nodes. The goal of multivariate network analysis is to utilize the topology of the multivariate network in conjunction with attribute data to derive new insights about a particular question of interest in domains such as biology, neuroscience and electrical engineering. Using the attribute data from MVN nodes and links, researchers and data analysts can discover new patterns and properties, leading to new ways of understanding and analyzing complex feature-rich networks such as social networks, brain connectivity networks, evolutionary hierarchy, and character relationships in film and literature.

Given the amount of data being generated every day due to advancements in sensors, networking, and computing [7], visualizing MVNs is a challenge because both the topology and the attributes of the nodes and links need to be displayed. Work done by Nobre et al. [19] and Lee et al. [14] have led to novel systems such as *Responsive Matrix Cells* [8] and *Juniper* [20]; however the tools and software available to implement these ideas for domain agnostic MVN analysis are limited. To address limitations of current MVN visualization tools and techniques, the *Collaborative Research: Framework: Software: HDR: Reproducible*

Visual Analysis of Multivariate Networks, an NSF funded research project led by Drs. Meyer and Lex is developing a novel system called *MultiNet* for performing visual analysis of multivariate networks. The three areas that the MultiNet project is investigating are [16]:

- Interactive, task-driven visualization of both the connectivity and attributes of networks;
- Reshaping the underlying network structure to bring the network into a shape that is well-suited to address analysis questions, and;
- Leveraging provenance data to support reproducibility, communication and integration in computational workflows.

As part of MultiNet, we propose *MultiAggr*, a proof of concept visual aggregation method for aggregating categorical node attributes for an adjacency matrix representation of an MVN. Our approach takes theoretical ideas from the MVN literature including supernodes, rollup and hierarchical-class based encodings, and combines these concepts to produce MultiAggr. Specifically, MultiAggr takes the MultiMatrix, an interactive adjacency matrix representation of an MVN, and produces the AggrMatrix, an interactive adjacency matrix representation of an MVN that allows users to perform detail-on-demand interactive analysis of the relationships of nodes in the MVN based on categorical attributes across a single axis. While this affordance is similar to Wattenberg's rollup technique [25], it is specifically adapted for the adjacency matrix visualization as part of the MultiMatrix application. By having the ability to perform visual aggregation, users of the MultiMatrix will be able to gain a high level overview of their MVN data and the ability to analyze patterns not visible when the data is not aggregated. In addition, our method allows users to analyze individual nodes from MultiMatrix and their connections to aggregated nodes and links in AggrMatrix which was not supported by *PivotGraph* [25]. Our contributions are the following:

- **MultiAggr**, a technique that implements a visual rollup specifically for an adjacency matrix representation of an MVN.
- **AggrMatrix**, an interactive adjacency matrix that enables users to visually analyze node attributes aggregated along a categorical attribute.

- A dual hierarchical visual encoding that represents the hierarchy between individual nodes and categorical node attribute groups.

- Superlinks, a single aggregated link between two supernodes.

- A user interface that supports detail-on-demand interaction of an aggregated MVN adjacency matrix representaton to visually study the relationships between nodes, links, supernodes and superlinks, and their attributes.

The rest of this thesis has the following structure. Chapter 2 presents the technical background and related works. Chapter 3 details our method MultiAggr. Chapter 4 presents an example of MultiAggr with the MultiMatrix using the *Les Miserables* character relationship dataset. Finally, we conclude in Chapter 5 with closing remarks and future work.

CHAPTER 2

BACKGROUND AND RELATED WORK

In this chapter, we describe approaches and techniques in visualization and data wrangling that have influenced the development of network visualization and network wrangling.

2.1 Network Visualization Layout

Nobre et al. [19] identified three types of visualization layouts for MVN visualization: Node-Link diagram layouts, tabular layouts, and implicit tree layouts. In this section we focus on defining terminology for node-link layouts and tabular layouts, specifically, the adjacency matrix layout. MultiNet visualizes MVNs using the Node-Link Diagram layout for the MultiLink application and the adjacency matrix layout for the MultiMatrix application. We also look at approaches to tabular visualization that can be applied to the adjacency matrix layout.

2.1.1 Node-Link Layout

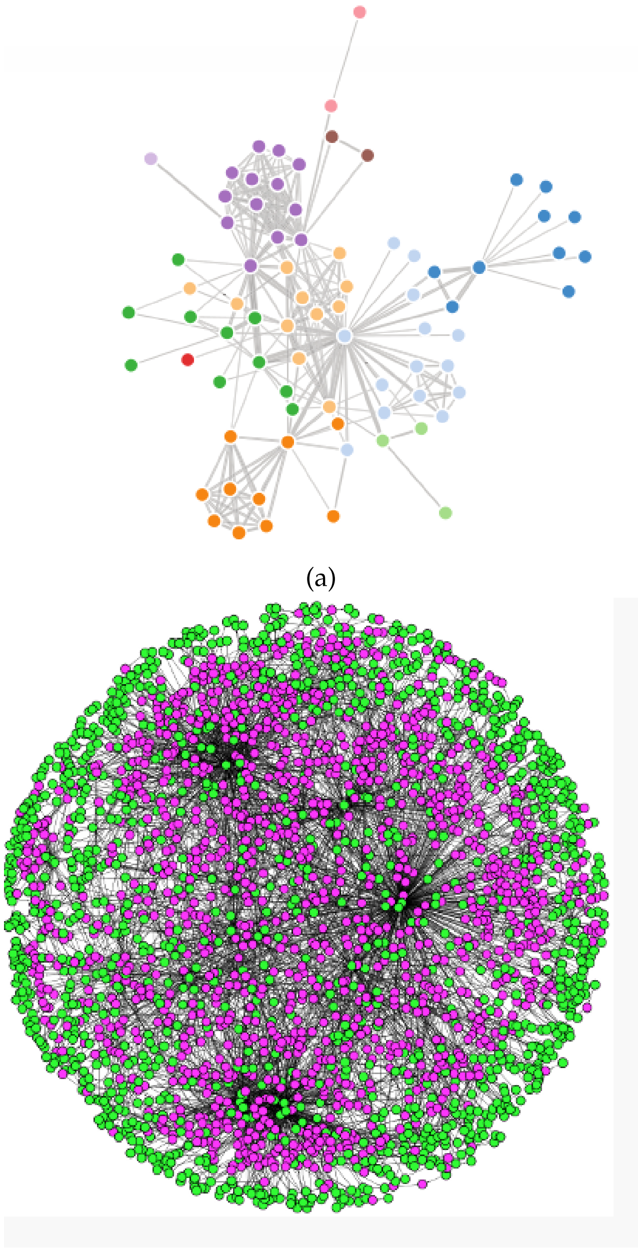
The Node-link diagram is one of the most common ways of representing a network. In this layout, nodes are represented as points and links are represented as lines [13] [18] [19] as shown in Figure 2.1a. This layout is easy to understand, particularly for people with minimal exposure to visualization and can be used to perform basic tasks such as finding the most popular person in a class [13] [18]. However, as the number of nodes and links increases, the Node-Link diagram layout degenerates into a *hairball*, where it becomes impossible to perform tasks such as identifying the topology of the network and connections between nodes [13] [18]as shown in Figure 2.1b. Recent approaches for scaling Node-Link layouts include coarsening large networks into smaller networks using wrangling operations [18], edge bundling techniques for grouping similar links into a single link [17], node filtering [23], edge lensing [9] and layout algorithms. While these approaches improve

Node-Link layouts and avoid the visual hairball for certain types of network data, they have not solved the problem for even larger classes of networks such as world airport flight connectivity. Nobre et al. [19] suggest a Node-Link layout for MVN visualization when the visual encodings can be easily understood by most users and when the MVN is sparse, that is, when there are few links and nodes. An alternative layout approach to this problem is the adjacency matrix layout discussed in the next section.

2.1.2 Adjacency Matrix Layout

The adjacency matrix is a type of *tabular layout*, where the nodes and/or links are represented as columns and/or rows of a table [19] as shown in Figure 2.2.

In this layout, a link between two nodes is encoded where the rows and columns intersect [18] [19]. The MultiMatrix application, as part of MultiNet, uses an adjacency matrix layout where the nodes are encoded as rows and columns of the matrix. The adjacency matrix representation is popular for topological feature analysis and node attribute visualization, but performs poorly for path analysis compared to the node-link layout [19]. Because it is a tabular layout, the adjacency matrix representation can be used to represent multiple attributes for nodes and edges concurrently, as well as, utilize a tabular visualization encoding and data wrangling operations such as filtering, sorting and aggregation [4] [6] [11] [19] [21]. The adjacency matrix layout requires quadratic screen space with respect to number of nodes, hence it limits the size of the network that can be visualized [19]. Despite this limitation, the adjacency matrix representation can visualize every possible connection for dense networks, which presents another solution to the visual hairball. Nobre et al. [19] recommend this representation for dense MVN data with rich nodes and/or edge attributes, and for all tasks except for path analysis. Given the visualization limitations for the adjacency matrix layout, interactive techniques and guidelines can be applied such that users can visualize particular groups of interest in their MVN data. In the next section we discuss principles of interaction useful for exploring large MVN data with visualization.



(a)

(b)

Figure 2.1: Node-Link Layouts. (a) Node-Link Diagram Layout for a small network by Gehlenborg et al. [5]. (b) The visual *hairball*, a large cluttered node-link layout for a dense network [13].

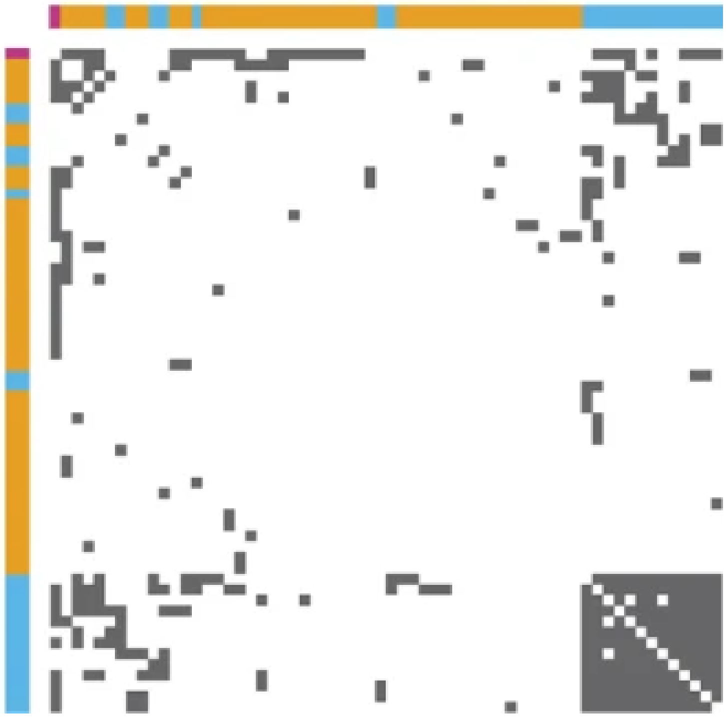


Figure 2.2: Adjacency Matrix Layout. An adjacency matrix layout for a network where the blue and bars indicate row and column nodes. The grey cells indicate a link between a row node and a column node [5].

2.2 Interaction in Visualization

Interaction is important for designing and building visualization tools as the complexity of a dataset increases [18]. Interaction is powerful in that it enables a user to explore information that is unavailable in a static visualization. However, according to Bostock [3] and Van Wijk [24], interaction also increases the complexity of a visualization and hides valuable information behind the endless number of settings a user can tweak to obtain insight through exploratory analysis. Shneiderman's [22]*Visual Information-Seeking Mantra*: Overview first, zoom, and filter, and then details on demand, provides a starting point for designing interactions for graph visualization tasks introduced by Lee et al. [14] and Nobre et al. [19]. For interactive MVN analysis, we focus on the first part of Shneiderman's Visual Information-Seeking Mantra.

Overview. The goal of overview is to give users a summary of the information in a dataset [18] [22]. This is particularly useful for dense networks and MVN data with the adjacency matrix layout because of the visualization space constraint. When designing overview interaction, the goal is to enable users to find areas of interest and guide them to perform further investigation in detail [18] [19]. To display multiple details of information at once with space constraints for overview, as in the case of visualizing nodes, links and their attributes for MVNs, Munzner suggests performing some kind of *reduction* operation that reshapes the data into a shape that can display all the information at once [18]. These operations are part of the data science pipeline stage called *Data Wrangling* [11] and include filtering, aggregating and sorting [2] [4] [11] [18]. In the next section we discuss data wrangling operations and how they can be applied to reshaping a MVN for visual analysis.

2.3 Data Wrangling

Data wrangling is the process of transforming raw data into a shape for data analysis. This process is tedious and usually takes up to 80% of the data science process [10] and the process usually requires using tools such as Python, R, or Excel [2] [10]. To perform visual analysis, data analysts use a series of reduction operations including but not limited to aggregation, sorting and filtering to identify areas for further detailed analysis [18]. Because of its tedious nature and the difficulty of maintaining a mental model of the data, researchers in the visualization community have been working on tools and systems to perform data wrangling operations and visualization concurrently [4] [11] [15] [21]. However, there are limited tools available for performing visual wrangling of networks. *Origraph* [2] and *PivotGraph* [25] are two recent systems that have been developed to address this problem and aid users to perform graph visualization tasks [14]. In the following section we focus on the aggregation network wrangling operation, aggregation tasks and visualizing an aggregated network.

2.3.1 Visualization and Network Wrangling for Aggregation

Earlier, we explained that the node-link layout and the adjacency matrix layout do not scale well as a network becomes more dense leading to visual clutter with the hairball and visualization space constraints. For MVN visualization, we also have to consider how to visualize attributes with nodes and links. In this work, we focus on visualizing wrangling categorical node attributes for MVN analysis. Categorical attributes are attributes that describe some category or property of an object. Examples of categorical attributes for a person would be hair color, eye color, and hometown. To reshape large MVNs for visual analysis using categorical node attributes we look at the tasks defined by Lee et al. [14] and Nobre et al. [19] and two systems *Origraph* [2] and *PivotGraph* [25]. These two techniques present *aggregation* as a solution for dense networks and MVNs.

2.3.2 Visual Aggregation

Lee et al. [14] identified two attribute based tasks for nodes: 1) *Find the nodes having a specific attribute value* and, 2) *Review the set of nodes*. These tasks are motivated by questions such as "Which people are your close friends in your social group?" and "How many of your friends attend the University of Utah?" These questions can be answered by creating a high level overview using aggregation for reduction [18] [19]. Aggregation is an example of *computing a derived value*, that is, given a set of data, compute a value representing the members of a group [1]. *PivotGraph* [25] focused on transforming node-link layouts of categorical node attributes using a novel algorithm called *rollup* for enabling users to gain a high level overview of their MVN data. The rollup algorithm takes an MVN and reshapes it to produce a new MVN where each node in the network represents an aggregate of all the nodes with a particular property. Similarly, rollup transforms the links into aggregated links between aggregated nodes which represent all the connections between two aggregated nodes. Bigelow et al. [2] defined these nodes as *supernodes* - the set of all nodes that belong to a particular group. A unique feature of PivotGraph [25] was the ability to use a control panel to choose the categorical attributes being displayed. This is an example of *details-on-demand* interaction to produce an *overview* of a particular group see many details at once [18] [19] [22]. In Figure 2.3 we show an example of *PivotGraph* for an aggregated Node-Link Diagram Layout. The links between two nodes represents aggregated links and the different colored points of varying radii indicate unique categorical groups with the size of the circle encoding the count for the aggregated values.

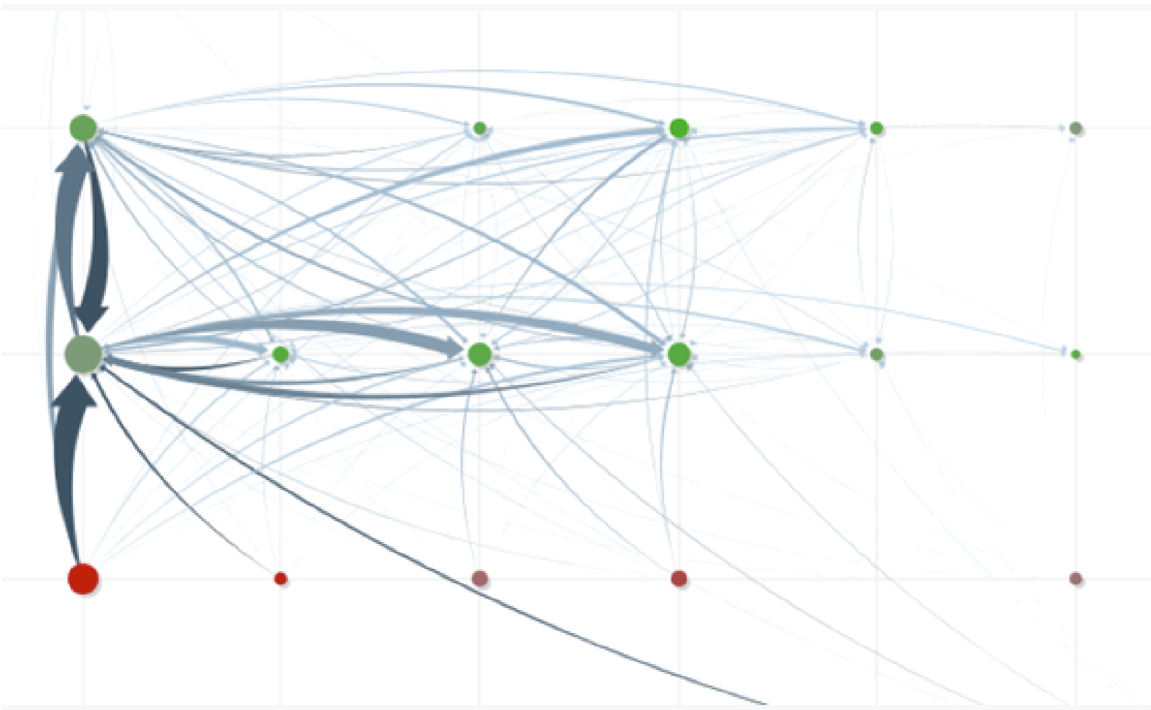


Figure 2.3: PivotGraph visualization of aggregated node-link-diagram layout.

This concludes the related works section and in the next chapter we discuss the *Multi-Aggr* Method.

CHAPTER 3

MULTIAGGR METHOD

In this chapter we detail our method *MultiAggr*, a visual aggregation technique for aggregating adjacency matrix representations of MVNs using categorical node attributes.

3.1 MultiMatrix

MultiAggr uses the MultiMatrix [16], an adjacency matrix representation of an MVN. The rows and columns of MultiMatrix represent the nodes in the MVN. To develop the interaction and visualization features for MultiAggr and AggrMatrix, we focused on refactoring MultiMatrix to have non-redundant visual encoding for connections [19] as well as features that enhance usability including toggling the matrix grid lines and supporting uniform labeling. The adjacency matrix layout supports both *directed networks*, where there is a link between a row node and a column node and vice versa, and *undirected networks*, where encodings are redundant above and below the diagonal [19]. MultiMatrix initially supported only directed networks which affected the count for aggregated nodes, so we built a toggle feature that enables users to toggle between directed and undirected MVNs. When a user toggles to a directed link representation, the matrix legend updates to reflect the new total number of connections between two nodes. MultiNet aims to support both directed/undirected MVNs so this feature enables users to choose the MVN directionality they need for their analysis.

Given Shneiderman's interaction principles [22] and Munzner's principles about interactive visualization design [18] we refactored the labels for nodes and built in support for toggling the gridlines of MultiMatrix. MultiNet does not specify the length or type of node names. For example, node names can range from numerical such as "1" to categorical node names of varying length of one more characters. MultiMatrix renders renders the node labels for rows and columns using SVG text, so to provide a consistent design for different labels we introduce a label cutoff width. MultiMatrix renders these labels up to the width and then adds ellipsis to indicate that the label is longer than the specified length. Using tooltips, a user can hover over a particular row or column node of interest to see the full length name. Usability is another important concern when designing visual interfaces and tools. We found that having the ability to toggle viewing the gridlines of the MultiMatrix enabled users to clearly see the links between row nodes and column nodes.

Given that the adjacency matrix is a type of tabular layout, we added in visual encodings to show different categorical node attributes using different colored rectangles inspired by Gratzl et al. [6], Furmanova et al. [4] and Rao et al's. [21] research for tabular multivariate visualization.

3.2 MultiAggr

MultiAggr can be described using Munzner's *What-Why-How* framework for designing a visualization method [18]. In the following sections we describe MultiAggr in terms of this framework.

3.2.1 Why

To support the first two capability goals of MultiNet, MultiAggr focuses on the issue of limited screen space for visualizing dense networks using the adjacency matrix layout. As discussed earlier, Nobre et. al [19] recommend using the adjacency matrix representation for most MVN tasks except for path analysis. At the time of this writing, MultiMatrix has limited the number of nodes that can be displayed to 1000 which limits the ability to visualize all the data information for greater than 1000 nodes. As described by Munzer, Shneiderman, Bigelow et. al and Wattenberg, we can use a reduction operation such as aggregation to display a high level overview of the data and collapse the matrix to a lower dimension to view all the MVN nodes, links and attributes [2] [18] [19] [22] [25]. Using the aggregated overview representation of an MVN, a user can perform detail-on-demand interaction to investigate the connections between aggregated nodes aggregated links and non-aggregated nodes and links. The ability to perform aggregation and visualization concurrently to produce an overview of important properties of an MVN is a powerful technique that allows users to visualize the aggregation pre-processing steps they are performing. For MultiAggr we focus on developing aggregation for categorical node attributes.

3.2.2 How

MultiAggr performs aggregation using categorical node attributes. Users of MultiMatrix select a categorical node attribute from the MultiMatrix Control Panel for how they would like to aggregate their MVN data using the adjacency matrix layout. Upon clicking the node attribute and enabling the aggregation functionality, the attribute table to the right of MultiMatrix updates to display the unique categories that belong to the attribute selected by the user using the colored rectangle visual encoding from *LineUp* [6]. When the user clicks the selected categorical node attribute in the attribute panel, a modified version of Wattenberg's Rollup Algorithm [25] designed specifically for the adjacency matrix layout is invoked, transforming MultiMatrix into the AggrMatrix. AggrMatrix is an aggregated representation of a MVN using the adjacency matrix layout encoding aggregated nodes on the rows and columns of the matrix using the supernode representation [2] and aggregated links for the intersection of row and column supernodes as a filled cell with a new color encoding. Next to the AggrMatrix buttons for controlling interactive data analysis of supernodes are generated. The AggrMatrix produces a new aggregated legend that visually indicates the reshaping of the MVN into a supernetwork and the number of connections between supernodes. The modified version of the Rollup Algorithm [25] collapses MVN nodes into a single supernode for a unique category and collapses links into *superlinks* an aggregated link between supernodes that we developed as part of MultiAggr inspired by supernodes [2]. To the right of AggrMatrix, the attribute table has been collapsed into a set of unique categories that belong to the selected categorical node attribute. One of the issues of *PivotGraph* was that it did not enable users to perform interactive data analysis using details-on-demand interactin [22] so we developed an algorithm for expanding and retracting AggrMatrix to view the connections between supernodes, superlinks, nodes and links. Upon expanding the AggrMatrix to visualize a group of interest, a hierarchical color encoding inspired by Kerzner et al.'s *Graffinity* system [12] is invoked to distinguish the difference between *supernodes* and their associated paths and *child nodes* and their associated path with dual color encoding legends in the MultiMatrix Control Panel.

3.2.3 What

The graph visualization tasks addressed by *MultiAggr* are providing a high level overview of dense MVN data and computing derived values for categorical node groups to show nodes with specific attributes [14] [18] [19] [22]. The goal of overview for the MVN adjacency matrix layout is to enable users to visualize as many details as possible about their MVN data [18] [19] [22]. By collapsing MultiMatrix to AggrMatrix for a single categorical node attribute, a user can display all the nodes in a dense MVN which provides a solution to the limited visualization space constraint described by Nobre et al. for the adjacency matrix layout [19]. In addition, aggregating by a single categorical node attribute versus multiple categorical node attributes takes advantage of pixel based displays by encoding a single value for a node attribute, thus making it easier to analyze a particular area of interest [5] Overview visualization by itself is not always effective because sometimes the insight it shares is difficult to interpret. The design choice to allow for details-on-demand interaction with expanding row supernodes using expanding/retraction controls for exploratory analysis was developed to allow exploration of all the nodes in a dense MVN. This allows for identifying particular patterns in a categorical node attribute group that were difficult to discover with only overview.

CHAPTER 4

EXAMPLE OF MULTIAGGR

In this chapter we demonstrate the different features of MultiAggr using the *Les Miserables* character relationship dataset. MultiAggr is a domain agnostic method and can be applied to any MVN whose nodes have categorical group attributes.

The *Les Miserables* character relationship dataset details the relationships between characters in the novel and the different scenes they interact in.

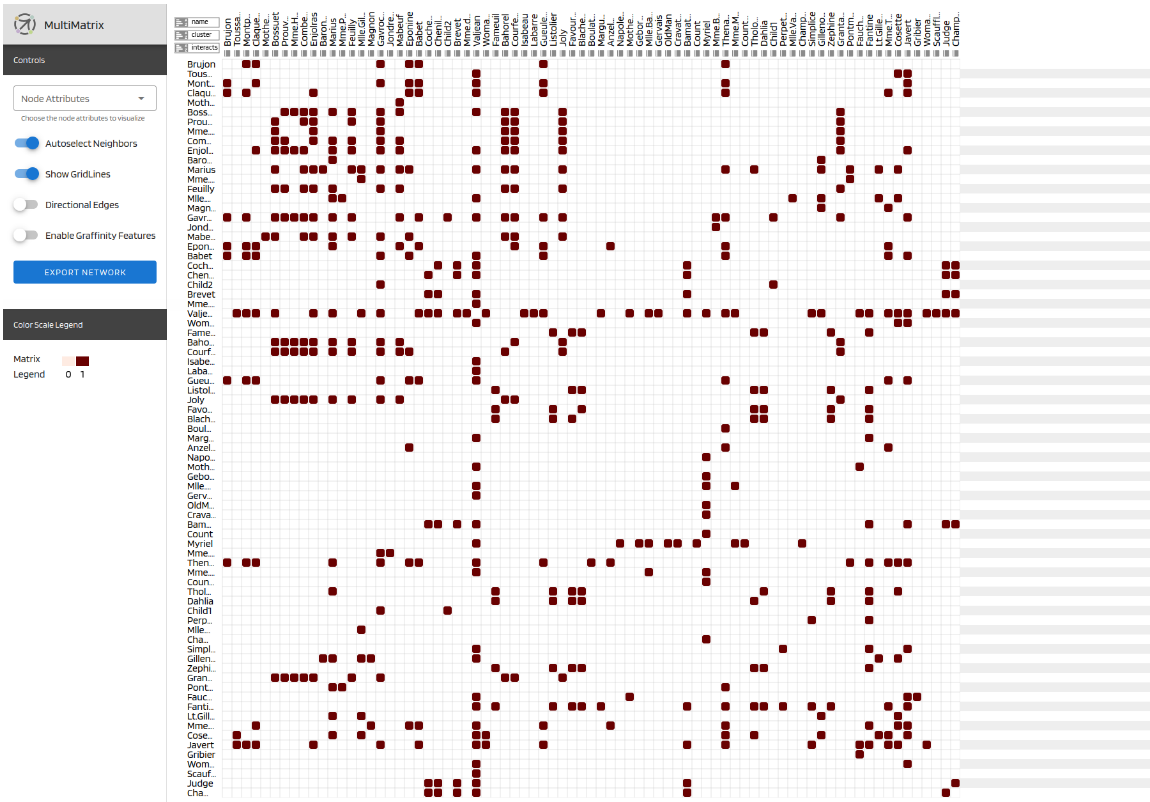


Figure 4.1: MultiMatrix: Non-aggregated adjacency matrix layout of *Les Miserables* character dataset.

We discovered based on expansion of the AggrMatrix that chapter 8 contains the most interactions between characters and is the climax of the novel. In chapter 8, Victor Hugo

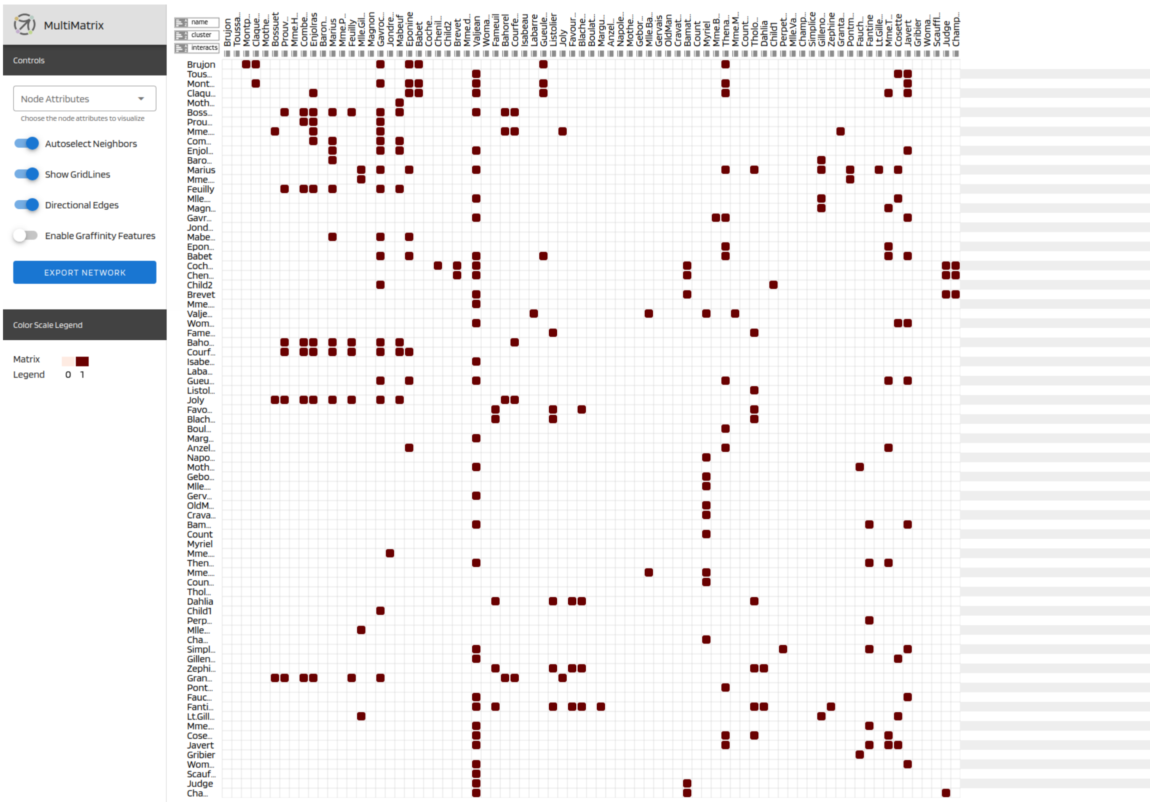


Figure 4.2: Directed Edges for MultiMatrix to indicate one way paths between nodes.

ties together all the previous threads about a robbery and how the characters from *Les Miserables* are connected to this event.

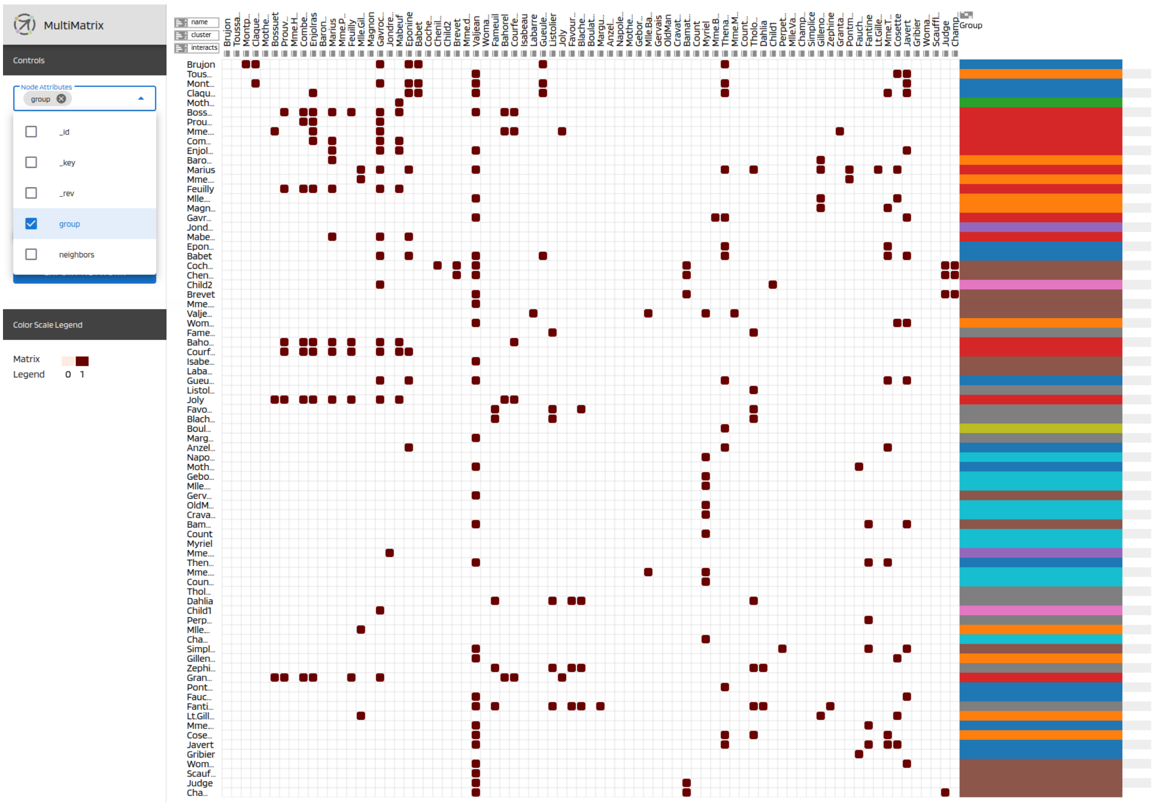


Figure 4.3: Selecting *group* categorical node attribute for *Les Miserables* data. On the right hand side of the matrix the different groups are now shown as different colored rectangles

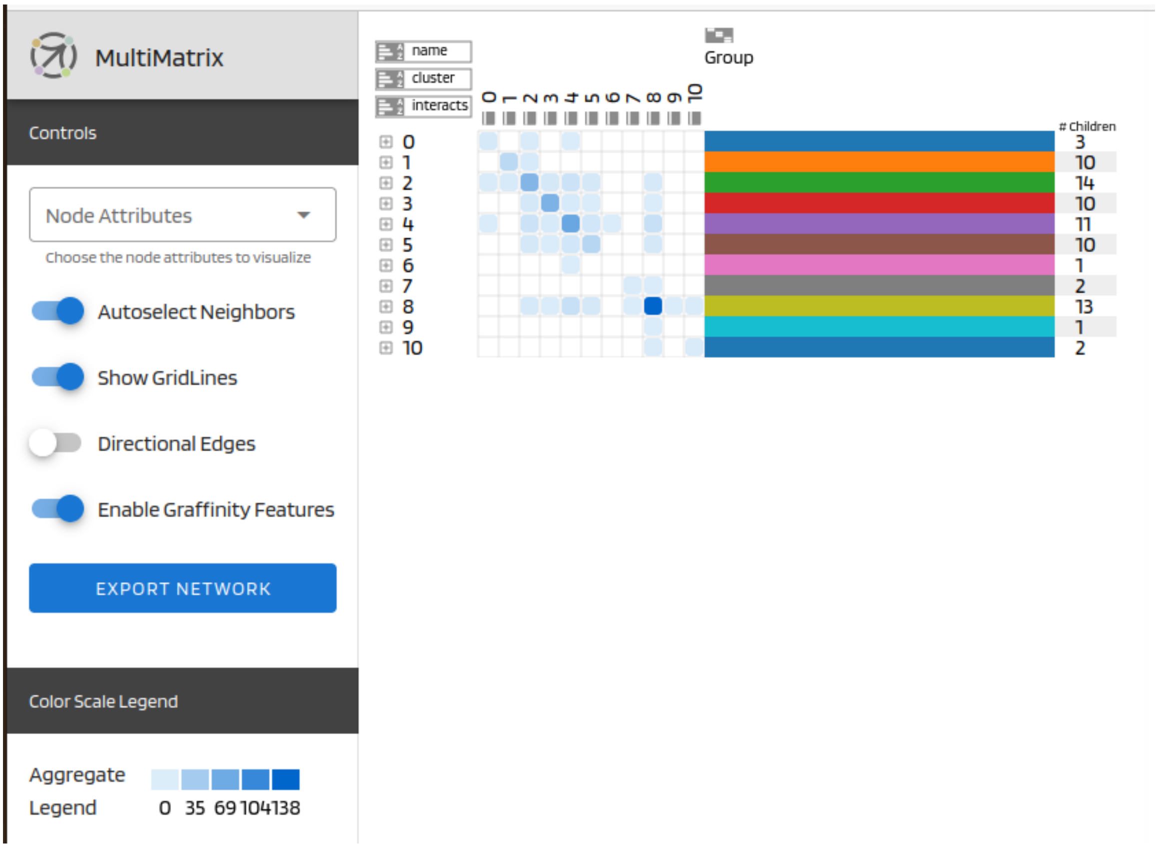


Figure 4.4: AggrMatrix: The aggregated adjacency matrix representation of a multivariate network featuring supernodes and superlinks. To the left of the supernode labels there are controls for expanding a supernode row to perform details-on-demand analysis to explore the child nodes that belong to a supernode group.

Figure 4.5: Expanding AggrMatrix to display the children nodes for chapter 8 supernode. This visualization features directed connections for nodes and supernodes and features the dual hierarchical color encoding as shown in the legend in the matrix control panel. In the attribute table, the attribute rectangle for chapter 8 has expanded to show that all the children nodes belong to chapter 8 with the green color encoding. Lastly the grid lines have been turned off to focus on where the row supernodes and children nodes intersect with the column supernodes and children nodes.

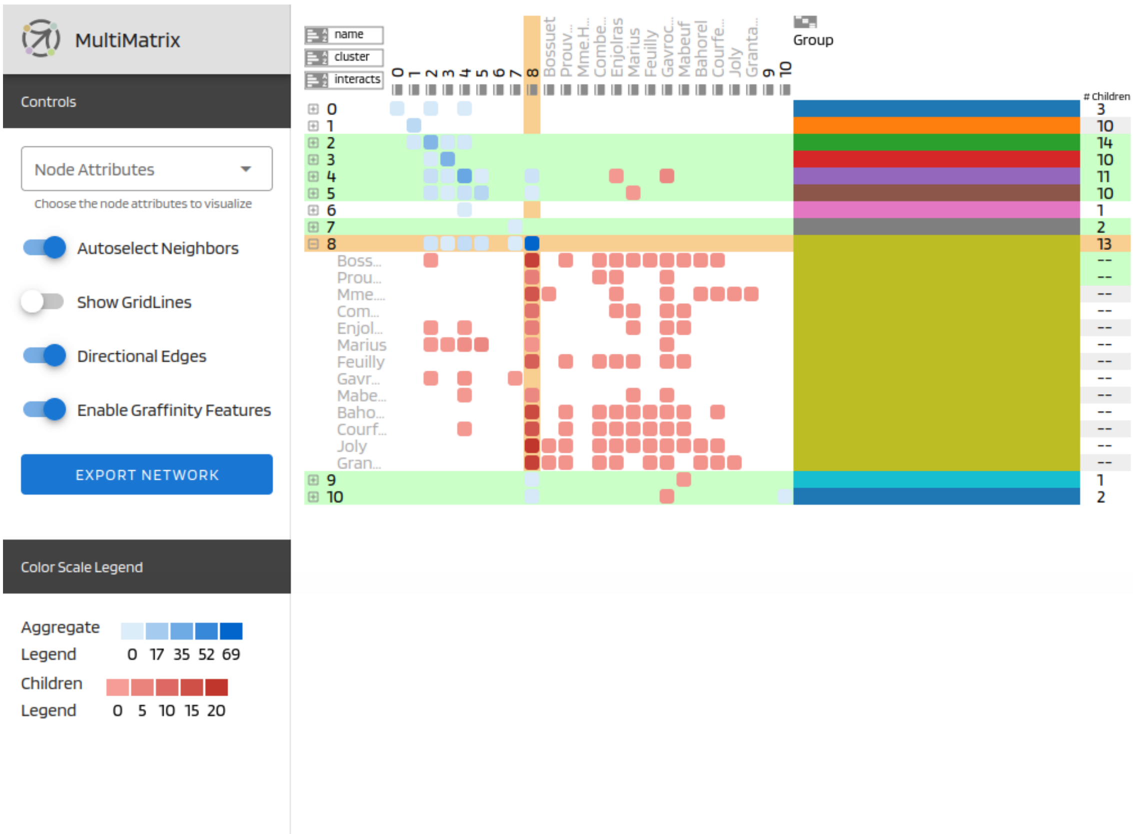


Figure 4.6: The expanded AggrMatrix featuring green highights to indicate all the nodes that the chapter 8 supernode intersects with to form a link.

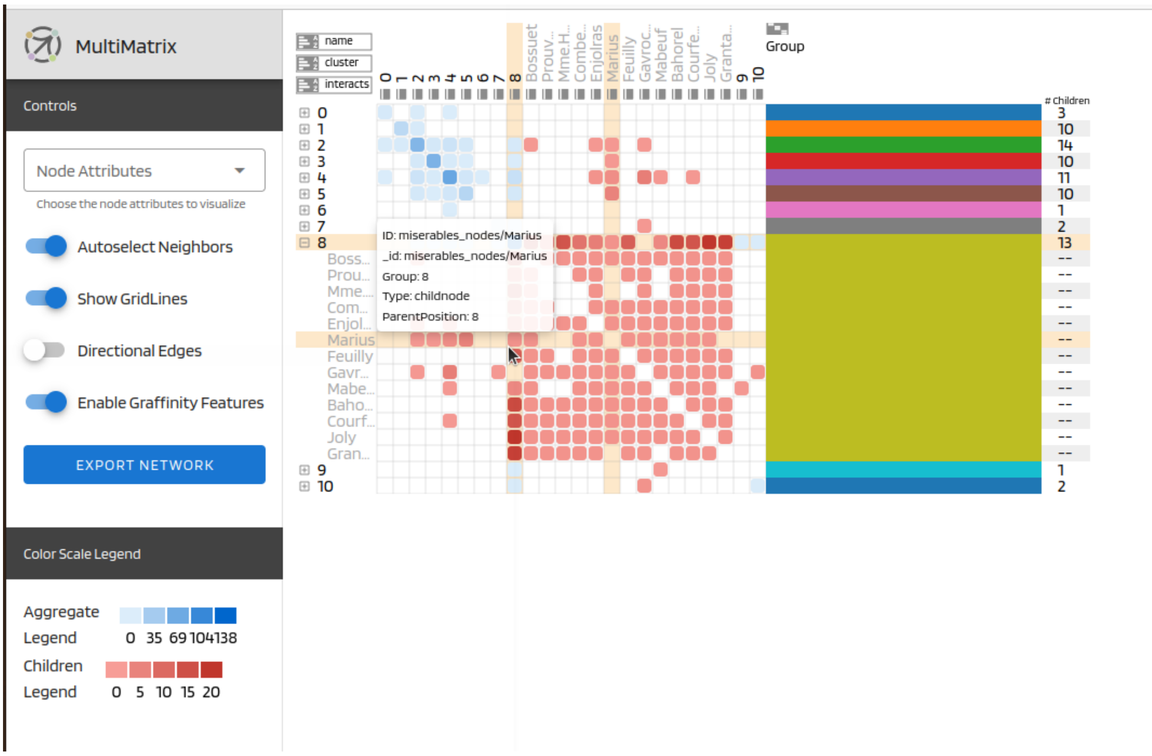


Figure 4.7: Tooltip highlight to show child node name for an expanded AggrMatrix. The tooltip contains information about the parent supernode and the type of the node, in this case a child node.

CHAPTER 5

CONCLUSION AND FUTURE WORK

By implementing visual aggregation for the adjacency matrix layout of an MVN using categorical node attributes, MultiAggr demonstrates a proof of concept technique for adjacency matrix MVN aggregation. In this project, we were able to address of the issues of the limitations of *PivotGraph* [25] and implement the tasks described by Nobre et al. [19] ans Lee et al. [14] for MVN visualization and aggregation using the adjacency matrix layout. MultiAggr is designed for aggregating domain agnostic MVN data using categorical node attributes across a single axis. As part of MultiMatrix, the MultiNet project hopes to build off of this work by supporting edge attribute visualization and aggregation, multiple node attribute aggregation and scaling for dense MVN data. Scalability is an ongoing area of research in MultiNet and we hope that MultiAggr will be able to perform on very large MVN datasets such as a global flight path MVN.

REFERENCES

[1] R. Amar, J. Eagan, and J. Stasko, *Low-level components of analytic activity in information visualization*, in IEEE symposium on information visualization, 2005. INFOVIS 2005., 2005, pp. 111–117. tex.organization: IEEE.

[2] A. Bigelow, C. Nobre, M. Meyer, and A. Lex, *Origraph: Interactive Network Wrangling*, in IEEE Conference on Visual Analytics Science and Technology (VAST), arXiv:1812.06337, 2019, pp. 81–92.

[3] M. Bostock, *10 years of open-source visualization*, Feb. 2021. Publisher: Observable.

[4] K. Furmanova, S. Gratzl, H. Stitz, T. Zichner, M. Jaresova, A. Lex, and M. Streit, *Taggle: Scalable Visualization of Tabular Data through Aggregation*, Information Visualization, 19 (2019), pp. 114–136. Publisher: Sage.

[5] N. Gehlenborg and B. Wong, *Networks*, 2012. Publisher: Nature Publishing Group.

[6] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit, *Lineup: Visual analysis of multi-attribute rankings*, IEEE transactions on visualization and computer graphics, 19 (2013), pp. 2277–2286. Publisher: IEEE.

[7] J. Heer, M. Bostock, and V. Ogievetsky, *A Tour through the Visualization Zoo: A survey of powerful visualization techniques, from the obvious to the obscure*, Queue, 8 (2010), pp. 20–30. Publisher: ACM New York, NY, USA.

[8] T. Horak, P. Berger, H. Schumann, R. Dachselt, and C. Tominski, *Responsive matrix cells: A Focus+ context approach for exploring and editing multivariate graphs*, arXiv preprint arXiv:2009.03385, (2020).

[9] C. Hurter, A. Telea, and O. Ersoy, *Moleview: An attribute and structure-based semantic lens for large element-based plots*, IEEE Transactions on Visualization and Computer Graphics, 17 (2011), pp. 2600–2609. Publisher: IEEE.

[10] S. Kandel, J. Heer, C. Plaisant, J. Kennedy, F. Van Ham, N. H. Riche, C. Weaver, B. Lee, D. Brodbeck, and P. Buono, *Research directions in data wrangling: Visualizations and transformations for usable and credible data*, Information Visualization, 10 (2011), pp. 271–288. Publisher: Sage Publications Sage UK: London, England.

[11] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer, *Wrangler: Interactive visual specification of data transformation scripts*, in Proceedings of the SIGCHI conference on human factors in computing systems, 2011, pp. 3363–3372.

[12] E. Kerzner, A. Lex, C. L. Sigulinsky, T. Urness, B. W. Jones, R. E. Marc, and M. Meyer, *Graffinity: Visualizing connectivity in large graphs*, Computer Graphics Forum (EuroVis), 36 (2017), pp. 251–260. tex.pmcid: PMC5821473.

[13] R. Kosara, *Graphs beyond the hairball*, Nov. 2014.

[14] B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry, *Task taxonomy for graph visualization*, in Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization, 2006, pp. 1–5.

[15] A. Lex, N. Gehlenborg, H. Strobelt, R. Vuillemot, and H. Pfister, *UpSet: Visualization of Intersecting Sets*, IEEE Transactions on Visualization and Computer Graphics (InfoVis), 20 (2014), pp. 1983–1992.

[16] A. Lex and M. Meyer, *Collaborative Research: Framework: Software: HDR: Reproducible Visual Analysis of Multivariate Networks with MultiNet*, Collaborative Research: Framework: Software: HDR: Reproducible Visual Analysis of Multivariate Networks with MultiNet, (2018).

[17] A. Lhuillier, C. Hurter, and A. Telea, *State of the art in edge and trail bundling techniques*, in Computer graphics forum, vol. 36, 2017, pp. 619–645. Number: 3 tex.organization: Wiley Online Library.

[18] T. Munzner, *Visualization analysis and design*, CRC press, 2014.

[19] C. Nobre, M. Meyer, M. Streit, and A. Lex, *The state of the art in visualizing multivariate networks*, in Computer graphics forum, vol. 38, 2019, pp. 807–832. Number: 3 tex.organization: Wiley Online Library.

[20] C. Nobre, M. Streit, and A. Lex, *Juniper: A tree+table approach to multivariate graph visualization*, IEEE Transactions on Visualization and Computer Graphics (InfoVis), 25 (2019), pp. 544–554. Publisher: IEEE.

[21] R. Rao and S. K. Card, *The table lens: merging graphical and symbolic representations in an interactive focus+ context visualization for tabular information*, in Proceedings of the SIGCHI conference on Human factors in computing systems, 1994, pp. 318–322.

[22] B. Shneiderman, *The eyes have it: A task by data type taxonomy for information visualizations*, in The craft of information visualization, Elsevier, 2003, pp. 364–371.

[23] M. E. Smoot, K. Ono, J. Ruscheinski, P.-L. Wang, and T. Ideker, *Cytoscape 2.8: new features for data integration and network visualization*, Bioinformatics, 27 (2011), pp. 431–432. Publisher: Oxford University Press.

[24] J. J. Van Wijk, *The value of visualization*, in VIS 05. IEEE visualization, 2005., 2005, pp. 79–86. tex.organization: IEEE.

[25] M. Wattenberg, *Visual exploration of multivariate graphs*, in Proceedings of the SIGCHI conference on Human Factors in computing systems, 2006, pp. 811–819.