

# Multivariate Network Exploration with JauntyNets

Iliir Jusufi, Andreas Kerren, and Björn Zimmer  
Department of Computer Science, ISOVIS Group  
Linnaeus University  
Vejdens Plats 7, 351 95 Växjö, Sweden

**Abstract**—The amount of data produced in the world every day implies a huge challenge in understanding and extracting knowledge from it. Much of this data is of relational nature, such as social networks, metabolic pathways, or links between software components. Traditionally, those networks are represented as node-link diagrams or matrix representations. They help us to understand the structure (topology) of the relational data. However in many real world data sets, additional (often multidimensional) attributes are attached to the network elements. One challenge is to show these attributes in context of the underlying network topology in order to support the user in further analyses. In this paper, we present a novel approach that extends traditional force-based graph layouts to create an attribute-driven layout. In addition, our prototype implementation supports interactive exploration by introducing clustering and multidimensional scaling into the analysis process.

**Keywords**—network visualization; multivariate networks; graph drawing; force-based layouts; attribute-driven layout; interaction; visual analytics;

## I. INTRODUCTION

The integration of multivariate data into complex networks and their visual analysis is one of the big challenges not only in visualization, but also in many application areas like bioinformatics [1], social sciences, or software engineering. Networks are usually modeled as graphs and represented with the help of node-link diagrams (or simple matrix representations). In social network analysis, for instance, each person can be represented as a node and the friendship between persons as an edge. A node could have a number of different attributes, such as name, age, gender, interests, etc. An edge might also have additional attributes, such as family relations (mother, father, or sibling) or a friendship weight (could be calculated based on the amount of activities, such as chatting or sharing the same interests between two persons). The most obvious challenge is now to show these attributes in combination with the underlying graph in order to allow the analyst to evaluate the meaning of the attributes in context of the network topology. Therefore, when faced with a data set that we want to better understand, some sort of insight about the attribute data and graph topology is necessary. Several questions could be important to answer in such analyses: Is the topology of the underlying network related to the values of particular attributes? Or, are specific network objects similar to each other in context of their

attribute values? Answers to these questions could help us understanding the structure of the dataset at hand and possibly lead us in formulating new interesting questions about the data.

In this work, we propose a novel approach that extends traditional force-based graph layouts to create a so-called *attribute-driven layout*. In addition, our prototype implementation, called JauntyNets, supports the interactive exploration by introducing clustering and multidimensional scaling into the analysis process. It should be noticed that we mainly focus on node attributes; edge attributes are restricted to be one-dimensional, i.e., we only consider traditional edge weights. The paper is structured as follows. Next in this section, we provide some basic definitions that we need to explain our approach. Section II gives a brief overview on related work. In Section III, our approach together with interaction possibilities is discussed. We conclude with a short discussion and possible future directions.

*Definitions:* In this paper, we distinguish between *graphs* and *multivariate networks*. A (simple) graph  $G = (V, E)$  consists of a finite set of vertices (or nodes)  $V$  and a set of edges  $E \subseteq \{(u, v) | u, v \in V; u \neq v\}$ . Whereas, a multivariate network  $N$  consists of an underlying graph  $G$  plus  $n$  additional attributes  $A = \{A_1, \dots, A_n\}$  that are attached to the nodes and/or edges. For node attributes,  $A_i$  represents a column in a table of attributes  $A = (a_{ji})$  ( $j = 1 \dots |V|; i = 1 \dots n$ ) and contains one attribute value per node (similar definition for edges). Thus,  $a^u = (a_{u1}, \dots, a_{un})$  describes all attribute values for node  $u$  given that there is no missing data.

Graph drawing algorithms compute the layout of nodes and edges mainly based on node-link diagrams [2] which play a fundamental role in network visualization. Applying graph layout algorithms can give valuable insight into the topology of a network if properly chosen and implemented. Whereas our work focuses on node-link diagrams, there are further possibilities to represent graphs, such as matrix representations [3] or hybridizations between both approaches [4].

Force-based layout techniques are in the focus of this paper, and a simple version using spring and electrical repulsion forces is introduced in the following. Here, the edges are modeled as springs, and the nodes can be considered as

charged particles that repel each other. For the x-component of the force on a node  $v$  the following holds (y-component analogous):

$$\sum_{(u,v) \in E} (sti_{uv}(d_{uv} - l_{uv}))\hat{x}_{uv} + \sum_{(u,v) \in V \times V} \frac{rep_{uv}}{d_{uv}^2}\hat{x}_{uv} \quad (1)$$

In this formula, the first sum represents the spring force between two nodes  $u$  and  $v$  connected with an edge and the second sum the repulsion force between  $v$  and other nodes.  $d_{uv}$  is the Euclidean distance between  $u$  and  $v$ ,  $l_{uv}$  is the zero-energy length of the spring between  $u$  and  $v$  (i.e., no force if  $d_{uv} = l_{uv}$ ),  $sti_{uv} \in [0, 1]$  is the stiffness of the spring between  $u$  and  $v$  (i.e., the larger this parameter the more the tendency for  $d_{uv}$  to be close to  $l_{uv}$ ), and finally  $rep_{uv}$  is the strength of the electrical repulsion between the two nodes. Note, that  $\hat{x}_{uv}$  denotes the unit vector of  $(x_v - x_u)$ .

## II. RELATED WORK

A traditional drawing algorithm will not solely solve our problem to visualize all elements of a multivariate network. There are several existing approaches that attempt to offer a solution to this challenge. The remainder of this section gives a brief introduction to these approaches. For a detailed survey, we refer the reader to Chapter 3 of the thesis [5].

*Multiple and Coordinated Views* combine several views and present them together. Usually, the views are coordinated by linking or brushing [6]. This well-known approach allows the use of the most powerful visualization techniques for each specific view and data set [7], [8]. The main drawback of this approach is that the displayed complex data is split, because of the spatial separation of the visual elements.

*Integrated Approaches* display attributes and the underlying graph in one view to provide a combined picture. “Integrated views can save space on a display and may decrease the time a user needs to find out relations; all data is displayed in one place.” [7]. One example is described in Borisjuk et al. [9] on the visualization of experimental data in relation of a metabolic network. This approach provides one view to all available information, but the embedding of the visualizations into the nodes causes the nodes to grow in size. This issue may affect the readability of the network due to the overlaps that may appear when the number of nodes and attributes is high [10]. Thus, it does not scale well. However, the problem of space usage and clutter introduced by such approaches can be avoided by using different focus + context techniques. An example of such an approach visualizes the attributes inside nodes by using a magic lens. Users can interactively build and combine various lenses by specifying different attributes and selecting suitable visual representations [11].

*Semantic Substrates* are used to avoid clutter in multivariate network visualizations. They are non-overlapping regions in which the node placement is based on the attributes of nodes [12]. One drawback of Semantic Substrates is that the underlying graph topology is not completely visible. A further related example is PivotGraph [13] which uses a grid-layout to show the relationship between (node) attributes and links. Another approach was presented by Pretorius and van Wijk [14]. They arrange edge labels in a list and place rectangular regions containing source and target nodes at each side. These regions are partitioned according to the node attributes and connected via straight lines with corresponding edge labels. Although the nodes “overlap”—regions can be contained in or contain other regions depending on the hierarchy level—they spatially remain in the same non-overlapping positions. Similar works have been presented by Archambault et al. in [15] and [16].

*Attribute-Driven Layouts* use the display of the network elements to present insight about attached multivariate data instead of visualizing the graph topology itself. While being similar to semantic substrates, this technique does not necessarily place the nodes into specific regions. Instead, it uses calculations based on node attributes to “steer” the placement of a node in the graph layout. Examples are GraphDice [17] or spherical Self-Organizing Maps [18].

*Hybrid Approaches* combine at least two of the previously discussed techniques. The most common combinations are multiple coordinated views with any of the integrated approaches. MobiVis [19] applies time charts to show temporal data in one view and the underlying network as node-link diagram in a separate view. Rohrschneider et al. [20] integrate additional attributes of a biological network inside the nodes and edges. They also use other visualization metaphors for creating multiple coordinated views to show time-related data of the network. Viau et al. [21] present another work where several different techniques are tightly integrated into one coherent hybrid approach.

## III. JAUNTYNETS

In this section, we discuss our visualization method. First, the creation of our sample data sets that are used for testing the prototype and taking the screenshots will be described. Afterwards, our visualization and interaction approach will be presented. A brief discussion of the most important implementation aspects will conclude this section.

### A. Sample Data

Our multivariate network data set D1 was created by using a set of text documents (a bunch of scientific papers and articles written by our research group). Another data set D2 was created by using the collection of papers from two different conferences. In both of these data sets, the nodes of the network represent the documents themselves. Each document owns a number of attributes that stand for the

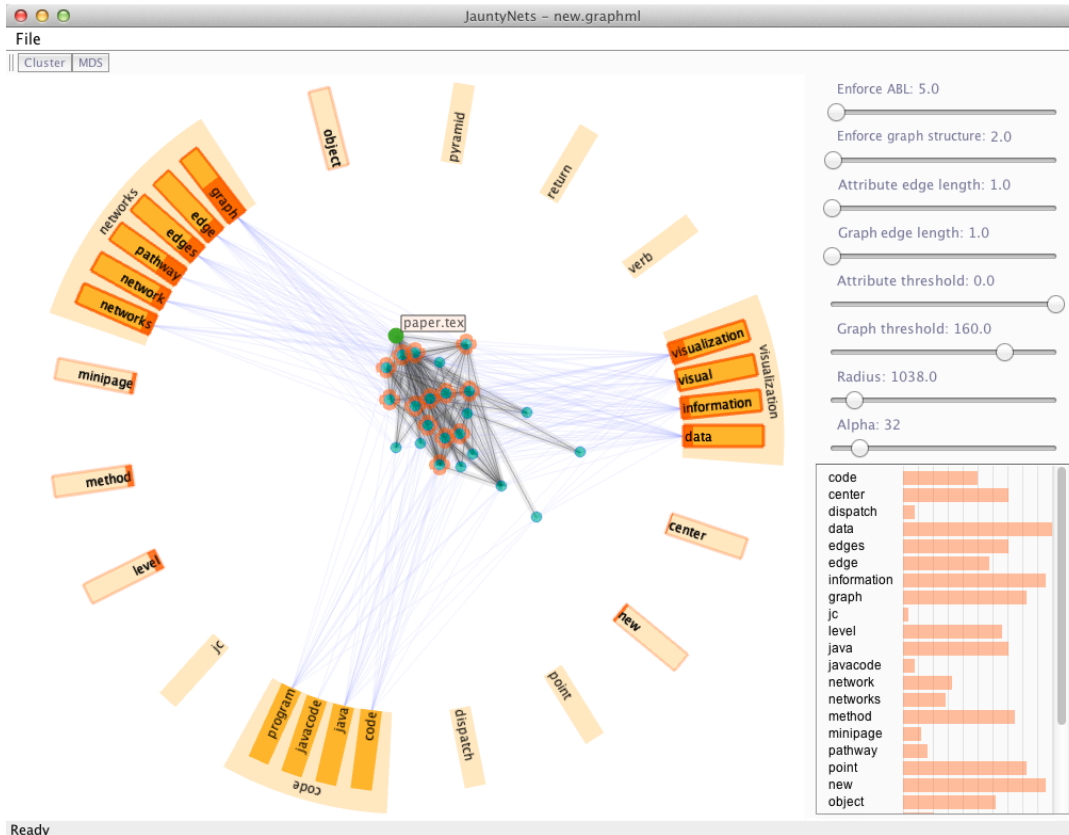


Figure 1. The graph view on the left shows three attribute groups (based on data set D1). The green node has been highlighted after a mouse-over action, and a tooltip displays the node label. The first-level neighbors of the selected node are shown with an orange halo. Attribute nodes act as bar charts for the highlighted node, giving insight into the values it has for each attribute. The faded attribute glyphs were disabled by the user.

occurrences of specific words within the document. Stop words, such as “a” or “the” were not included. The similarity between the documents is calculated based on the co-occurrence of the attribute set among the documents and is represented as an undirected edge between the corresponding nodes. The weight of an edge corresponds to the degree of similarity calculated as the sum of the minimum values of the attributes. This input data was filtered by setting a specific threshold for the minimum occurrence of words and for minimum edge weights.

Additionally, a Jigsaw data set D3 containing metadata for every IEEE InfoVis and VAST conference paper from 1995 to 2011 was visualized by using our approach as well [22]. Here, the links between nodes represent co-authorship, i.e., if two papers share an author, then their node representations are connected with an edge. The weight of an edge corresponds to the number of shared authors. *Concept terms* are metadata that were identified in titles and abstracts of the papers. These terms are used as attributes for D3. The conference where the paper has been published serves as attribute as well. Keep in mind that for this data set, the individual attributes only have two possible values: one and zero.

## B. Approach

Our idea could be described as a *hybrid approach* that combines *multiple coordinated views* with *attribute-driven layout*. The exploration of the networks is carried out through a number of available interactions. Users can change different parameters of our visualization tool in order to get more insight into the data set. Additionally, data mining techniques, such as clustering and multidimensional scaling, were used to facilitate the visual analyses of multivariate networks.

*Extending Traditional Force-based Approaches:* Our main contribution in this paper is the extension of a force-based approach as described in the introduction, so that it produces an attribute-driven layout embedded into JauntyNets for the interactive visual analysis of multivariate networks. The input graph  $G$  is placed in the center of the main view (cf. Figure 1). Here, the natural edge length  $l_{uv}$  is inversely proportional to the edge weight, i.e., the larger the weight, the closer the nodes are positioned. This way, more similar documents are placed closely together. The attribute nodes  $A_i$  are placed on a circle around the network to be visualized. The radius of the circle is interactively adjustable by using a slider. Technically, attributes are nodes

---

**Algorithm 1** Building Network Structure

---

**Input:** network with attributes  $N = (G, A)$  where  $G = (V, E)$ ;

**Output:** extend  $N$  so that  $V \leftarrow A$

```
// Add attributes to vertices and mark them as attribute
vertices.
1: for all  $a$  in  $A$  do
    // Create attribute vertex  $v_a \in V_a$  such that  $V_a \subseteq V$ 
2:    $v_a \leftarrow a$ ;
3:    $V.add(v_a)$ ;

    // Create edges from attribute vertices to graph ver-
    tices and compute edge weights based on attribute
    values.
4:   for all  $v$  in  $N$  do
5:     if  $v.attribute(v_a).hasValue()$  then
6:        $e = \text{new Edge}(v, v_a)$ ;
7:        $e.weight(v.attValue(v_a))$ ; // Assign attribute
       value to the edge weight.
8:        $E.add(e)$ ;
9:     end if
10:  end for
11: end for
```

---

---

**Algorithm 2** Layout Algorithm

---

**Input:** network  $N$  such that  $V_a \subseteq V$  // cp. Algorithm 1.

**Output:** vertex placement  $p = (p_v)_{v \in V}$

```
// Perform initial placement of vertices.
1: placeAttributeNodesOnCircle( $V_a$ );
   // Place vertices randomly inside the circle.
2: placeNodesInsideCircle( $V - V_a$ );

   // Run force-based layout algorithm. Do not apply forces
   to attribute nodes positions.
3: loop
4:   for all  $v$  in  $(V - V_a)$  do
5:      $F_v \leftarrow F_{rep(v)} + F_{spring(v)}$ ; // cp. Eq. (1)
     // If a node is pushed over the circle radius, apply
     force towards circle center.
6:     if  $p_v.isOutsideRadius()$  then
7:        $F_v \leftarrow F_v + F_{gravity(v)}$ ;
8:     end if
9:   end for

10: for all  $v$  in  $(V - V_a)$  do
11:    $p_v \leftarrow p_v + \epsilon \cdot F_v$ ;
12: end for
13: end loop
```

---

as well, movable on the circle only by user interaction; they are called *attribute nodes* ( $V_a$ ) in the remainder of this paper. The main difference to a traditional force-based layout method is that we do not apply forces on attribute nodes (therefore, they are not moved to new positions by the incremental layout algorithm). Additionally, we have embedded another gravitational force that is triggered when a normal node crosses the circle perimeter in order to prevent it from leaving the circle due to the repulsion forces between the nodes (so-called *position constraint*). Only nodes that have attribute values over a specific threshold are linked to attribute nodes via so-called *node-to-attribute edges*. Similarly, only edges with weights over a user-defined threshold are shown. All these thresholds, including the stiffness  $sti_{uv}$  of normal edges and node-to-attribute edges, can be set up by using the corresponding sliders. The “Enforce ABL” (Attribute Based Layout) slider is used to control the stiffness of node-to-attribute edges and its function is to affect node positions based on their attribute values. The “Enforce graph structure” slider controls the stiffness of the graph (node-to-node) edges and is used to enforce the graph topology. By in- and/or decreasing these two values, users can choose to preserve graph topology, gain insight into attribute values or try to balance both. Please compare Algorithm 1 to see how the attributes are *transformed* and appended to the underlying graph. Algorithm 2 shows the details of the extension compared to the work of Eades [23]. As previously mentioned, the positions of the attribute nodes  $V_a$  are not affected by our algorithm. Therefore, they are excluded in line 4 from the loop and no further forces are being applied on them. Another extension of the algorithm happens in lines 6-8 where another force  $F_{gravity(v)}$  is being applied that pulls the nodes towards the center of the circle in case they tend to move away as a result of interactions with other nodes.

Figure 2 shows the initial view after sample network D1 has been loaded to the system. Notice that nodes are highly connected to each other, because the sample network represents a selection of 24 research papers in total written by members of our research group. Thus, all of these documents are fairly related to each other. At this point, users might want to perform a number of interactions with JauntyNets in order to analyze the data set further.

*Visualization and Interaction:* As already mentioned in Subsection III-A, edges can be filtered out by using a slider if their weight is below a certain threshold. The weights are represented by the thickness of the edges. A user can drag and move the attribute nodes to control their position on the circle. Inner nodes will then automatically move based on the attribute position they are linked to. Users can disable certain attributes by first selecting the attribute node(s) and pressing the key “A”, in case they consider them as unnecessary for the analysis process. Furthermore, one can create groups of attributes that share some kind of

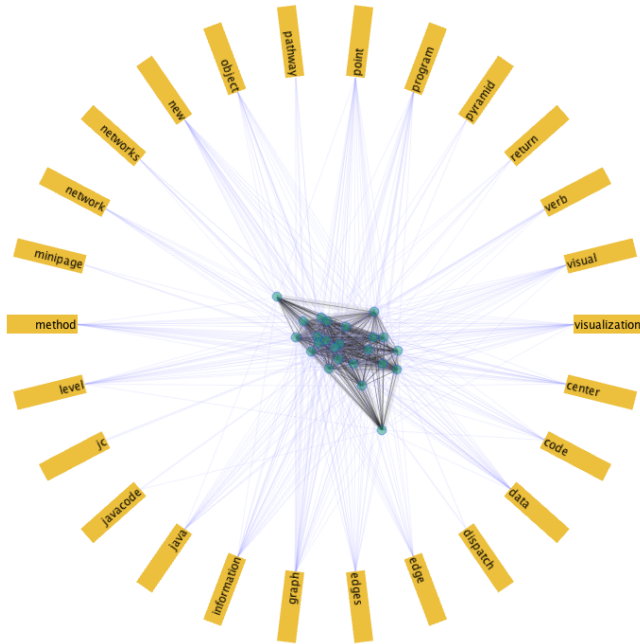


Figure 2. This screenshot shows the initial visualization after the data D1 has been loaded. The orange blocks placed on a circle represent the attributes of the network. Green nodes in the middle represent the text documents.

meaning, i.e., they can be subsumed under a superordinate concept. Document nodes that correlate with a group of attributes will then move towards that group. Of course, the edges between the documents themselves will affect the positioning. Our tool also supports the grabbing of an entire group and moving it around the circle, instead of moving the desired members individually.

Similar interaction steps have been performed to our initial data set D1 as shown in Figure 1. Edges of the graph that show weak relations among different documents have been filtered out. The stiffness parameter  $sti_{uv}$  for node-to-attribute edges is increased by using a slider. Nodes which have a higher value of a certain attribute are thus moved towards the corresponding attribute. Next, three attribute groups have been created in our screenshot example: *visualization*, *networks*, and *code*. To create a group, users can select multiple attributes with a right-click and press “G” on the keyboard. To add more attributes to a specific group, one can move the attribute inside the group region and press “G” again. The group is initially labeled with the name of the first attribute added to the group. Later, the name can be changed through a context menu. The final groups should be distributed evenly around the circle. In our concrete sample data set in Figure 1, we see that the node positions with respect to the groups correspond to the major topics described in the input documents. Thus, we have gained insight into the content of these documents.

If a user places the mouse cursor over a node, JauntyNets highlights the corresponding node and displays a tooltip with the node label as displayed in the center of Figure 1. The attribute nodes connected to the selected node mimic bar charts showing the concrete values of the attributes. The direct neighbors of the node are highlighted too. Similarly, if the user places a cursor over a specific attribute node, all nodes that have a node-to-attribute edge to the attribute will be highlighted. This way, users can see which documents contain that particular attribute (keyword in this case). Keep in mind that only the attributes that pass a certain threshold are considered. It is also possible to change the visualization parameters, such as the radius of the attribute circle or the transparency value of the attribute-to-node edges, as well as to zoom-in/-out or to pan the view.

Figure 3 represents the visualization of the Jigsaw data set D3. Only a predefined number of the concept terms have been preloaded. These terms are related to the field of “geovisualization” and as such have been grouped together, as seen in the lower part of the screenshot. Two other attributes stand for the conferences where the papers have been published. Blue nodes represent InfoVis, while the red ones represent VAST papers. At this point users can play with the sliders. They might choose to enforce the graph topology, or they might want to enforce the attribute values in which case the nodes will be moved towards attributes for those they have the highest values. Figure 3 represents balanced parameters, i.e., neither topology nor attribute values have a higher coefficient. This example gives us an overview of the paper distribution between the two conferences. Certain cliques can be noticed immediately. Such network structures usually mean that one specific author has written all those papers, or there is a group of authors who often cooperate. We see that one such clique is placed relatively close to our “geovisualization” group (Figure 3 marked by “A”). Therefore, we may decide to investigate further. Upon going through the papers from that clique, we identify one common author (MacEachren). We also discover that only one of those papers is not related to “geovisualization” as none of the attributes of the “geovisualization” group is connected to it. This can be visualized with the mouse-over function on either attributes or papers as described earlier. In contrast to this, another clique marked by label “B” only comprise papers published at the InfoVis conference. Upon further investigation, we notice more than one common author. By filtering out those edges that have less than two authors (edge weight), the analyst is able to see that the clique remains again although two nodes have been separated. Therefore, we can conclude that these authors (Dykes, Slingsby and Wood) have worked closely together, and published their work at the InfoVis conference (Figure 4). However, half of their papers are not related to “geovisualization”. This shows that the authors of clique “B” are not mainly focused on one domain of information visualization, in contrast to the author

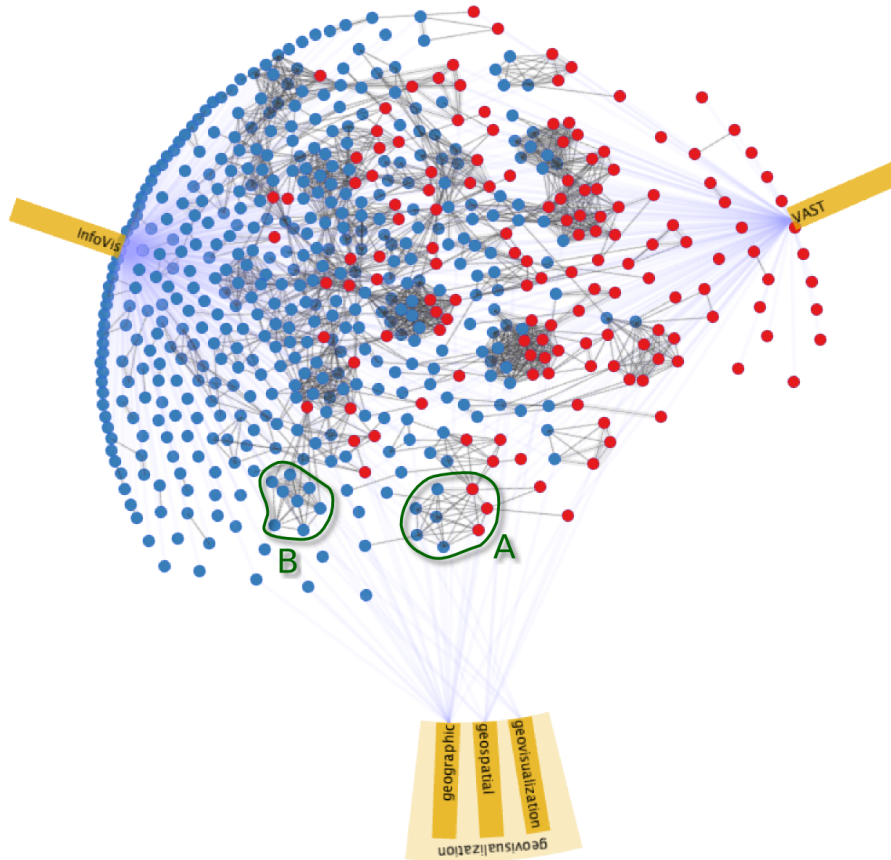


Figure 3. The screenshot displays papers from InfoVis and VAST conference (data set D3). Two cliques marked with A and B are interesting as they are related to the created attribute groups.

of the clique “A”. With such an exploratory approach, we can discover if certain parts of the network structure are related to particular network attributes.

As discussed in Section II, there are different approaches of attribute-driven layout for multivariate networks. However, most of these tools affect the network layout drastically. Our aim is to preserve the graph topology as much as possible, i.e., to interfere less with the traditional force-based layout algorithms. However, we are aware that this approach alone would not produce extensive analysis results. Therefore, several other (standard) views and techniques are added to enable a more in depth exploration of the data.

*Clustering:* In order to further facilitate the exploration process, a possibility to cluster the nodes has been added to JauntyNets. Figure 5 shows a more complex network (data set D2) with 421 nodes and 55 attributes where the nodes have been clustered using a k-means clustering algorithm [24]. This data set consists of various articles published in the VisWeek<sup>1</sup> and INTERACT<sup>2</sup> conference

proceedings from 2009 to 2010. To initiate the cluster process, a user just needs to choose “Cluster” from the toolbar menu and has to select the desired number of clusters. The clustering of the nodes is done based on their attribute values. We use an external library to perform the clustering (cp. Subsection III-C) which offers an option to choose between several clustering algorithms. In our case, three clusters have been created (see Figure 5). The color coding of the clusters has been done using ColorBrewer’s categorical color map [25]. Note that those attribute groups (i.e., “interaction” and “mobile”) comprising HCI-related keywords attract mainly nodes from the *blue* cluster, which is primarily consisting of nodes (papers) from the INTERACT conference that focuses on HCI. The clustering is additionally used for automatic attribute group creations. In this case, attributes are clustered based on the node values for a particular attribute. This speeds up the attribute group creation process. Additionally, it could give insight into how different attributes relate to each other for specific data sets. However, users might want to shuffle the attributes between the groups to achieve optimal grouping.

<sup>1</sup><http://visweek.org/>

<sup>2</sup><http://www.interact2013.org/>

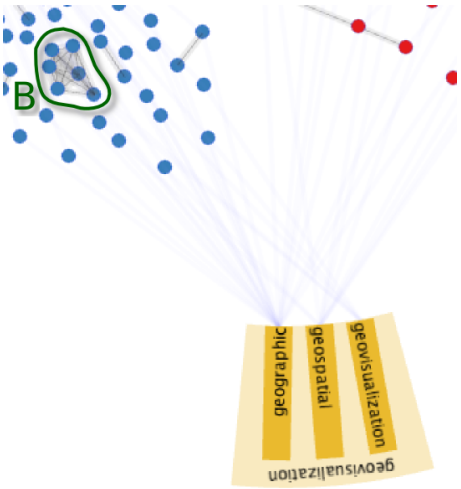


Figure 4. A cut-out of the data represented in Figure 3. It shows that clique B is still highly connected even after filtering out all edges with weight 1.

*Further Coordinated Views:* As mentioned at the beginning of this section, our approach uses multiple coordinated views to support the exploration of multivariate networks. The first additional view shows the number of nodes that a certain attribute has an edge to. The view changes accordingly, based on the specified threshold, and uses a simple bar chart metaphor as shown in Figure 1, down to the right. It is helpful in such cases when we have attribute outliers with too many connections, i.e., such attributes might be too general to be considered useful for exploration with our approach. Thus, we might try to disable them to achieve a better positional result as done with “display” in the interaction group of Figure 5, for instance.

Another view displays a multidimensional scaling (MDS) visualization. Here, multivariate network nodes have been projected into the two-dimensional plane based on their attribute values. Similarity between nodes is represented as distance, i.e., the closer two nodes are, the more similar their content is. Figure 6 shows the projection based on the network in Figure 5. This view is related to the graph view as it shows similar interaction possibilities in terms of highlighting, zooming and panning. A selected node in one view will be highlighted in the other view as well. Thus, the user can track the position of the nodes in both views. It is interesting to notice that the clustering results complement the spatial positioning of the MDS. We perceive a dense area of the nodes in the upper-left corner belonging to the *blue* cluster. With further interactions, it can be discovered that these nodes (articles) mainly belong to the INTERACT proceedings. This finding is in accordance with the results from Figure 5 as described earlier. Therefore, the MDS view could be helpful for answering questions on the similarity of the network objects in context to their attribute values.

### C. Implementation

Our prototype has been implemented in Java together with the *Processing* [26] graphic library using *OpenGL*. For implementing the multidimensional scaling feature, we have used the *MDSJ* library [27]. Thus, we only had to compute the dissimilarity matrix as input. Clustering has been performed by using the *trickl-cluster* library. This library offers several clustering algorithms that could be used alternatively [28]. Additionally, the *giCentre Utilities* [29] libraries provide color maps for our clustering based on *ColorBrewer* [25]. We extended the traditional force-based layout algorithm by including static nodes as attributes (Algorithm 1), edge weights, and position constraints as presented in Algorithm 2.

## IV. CONCLUSION AND FUTURE WORK

Multivariate relational data sets are common in several application domains. They are important foundations to explore different processes, such as protein-protein interactions in the life sciences or interpersonal activities in social sciences. Although these fields may appear diverse, one can generalize problems pertaining to multivariate networks across different domains. With this in mind, we presented an approach that enables the interactive exploration of such data.

Our main contribution is the extension of a force-based graph layout approach to act as an attribute-based layout system. With this extension, we managed to affect the graph topology so that each node’s position is related to the corresponding attributes’ values. By doing so, users are able to gain insight into the overall multivariate network structure. The interaction techniques and the visualization approach allow users to focus on the topology, or attribute values and even to achieve a balance between both by tweaking different parameter values thus exploring the data interactively. Furthermore, we added clustering and multidimensional scaling features to support the analysis process [30].

*Future Work:* Adding more visualization views and techniques that focus on certain types of data visualization will be an important future work. For instance, parallel coordinates could be used for a separate attribute visualization which could be helpful for filtering the data set.

There are cases when some nodes are unconnected from the network as a consequence of filtering. Currently, these nodes are positioned inside the circle and their positioning is random. To avoid any ambiguity regarding the positioning of these nodes, they should be placed outside the circle in a (possibly separated) region. Additionally, inactive attribute nodes create visual overload. This problem could be solved in a similar manner by placing them inside of a separate region of the screen. Users can reactivate them again if necessary. We also plan to perform an evaluation with different domain experts, which will help us to improve the existing approach.

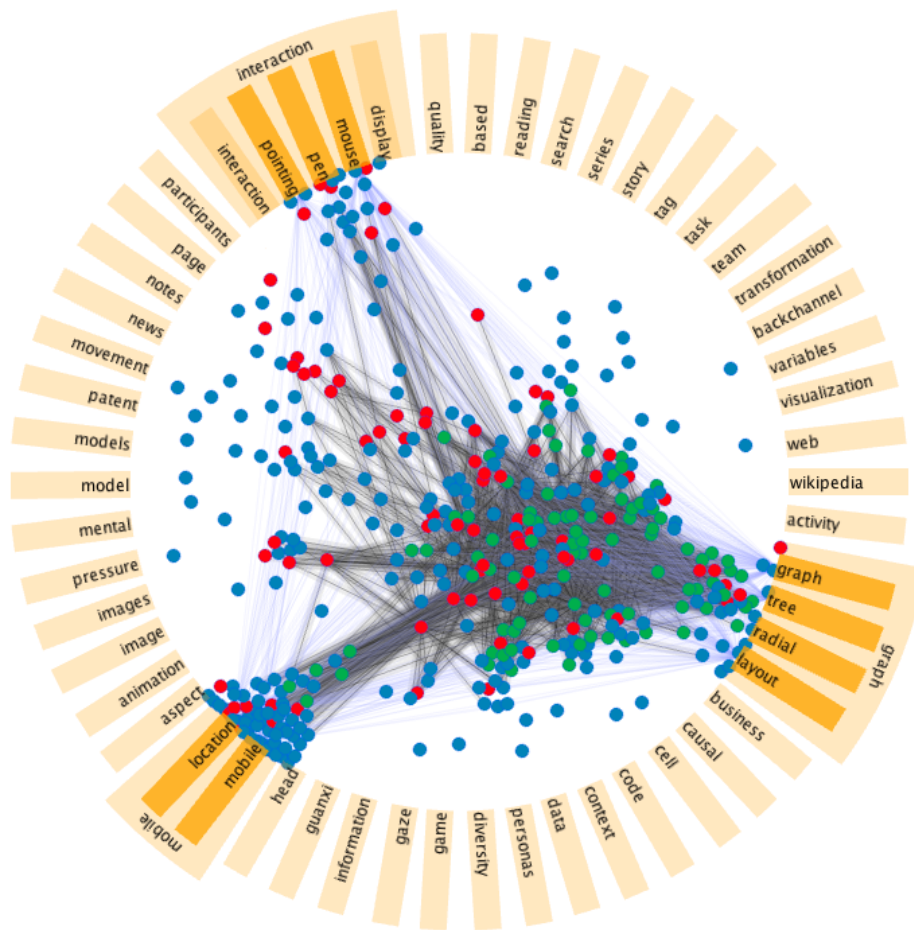


Figure 5. The screenshot displays a network of 421 nodes with 55 attributes (data set D2). The nodes are colored differently, because they have been clustered based on their attribute values.

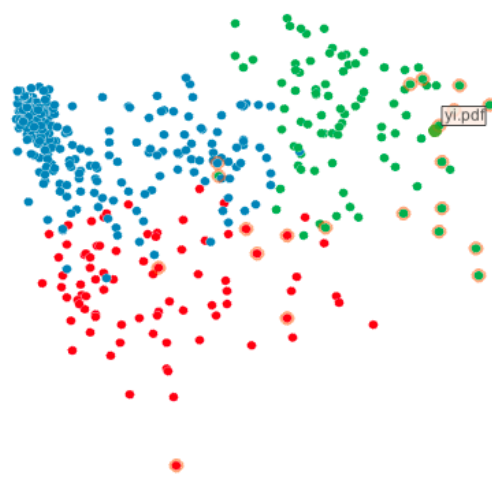


Figure 6. MDS view (data set D2). The highlighted node shows a tooltip with the node label “yi.pdf”, and the first-level neighbors are displayed with an orange halo. The node colors reflect the results of a previous clustering step.



## REFERENCES

- [1] M. Albrecht, A. Kerren, K. Klein, O. Kohlbacher, P. Mutzel, W. Paul, F. Schreiber, and M. Wybrow, "On open problems in biological network visualization," in *Proc. International Symposium on Graph Drawing (GD '09)*, ser. LNCS, vol. 5849. Springer, 2010, pp. 256–267.
- [2] C. Ware, *Information Visualization: Perception for Design*, 2nd ed. Morgan Kaufmann, 2004.
- [3] J. Abello and F. van Ham, "Matrix zoom: A visual interface to semi-external graphs," in *Proceedings of the IEEE Symposium on Information Visualization*. Los Alamitos, CA, USA: IEEE Computer Society, 2004, pp. 183–190.
- [4] N. Henry, J.-D. Fekete, and M. J. McGuffin, "Nodetrix: a hybrid visualization of social networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, pp. 1302–1309, 2007.
- [5] I. Jusufi, "Towards the visualization of multivariate biochemical networks," Licentiate Thesis, Linnaeus University, 2012.
- [6] R. Spence, *Information Visualization: Design for Interaction*, 2nd ed. Prentice Hall, 2007.
- [7] C. Görg, M. Pohl, E. Qeli, and K. Xu, "Visual Representations," in *Human-Centered Visualization Environments*, ser. LNCS Tutorial 4417, A. Kerren, A. Ebert, and J. Meyer, Eds. Springer, 2007, pp. 163–230.
- [8] J. C. Roberts, "Exploratory visualization with multiple linked views," in *Exploring Geovisualization*, A. MacEachren, M.-J. Kraak, and J. Dykes, Eds. Elsevier, December 2004. [Online]. Available: <http://www.cs.kent.ac.uk/pubs/2004/1822>
- [9] L. Borisjuk, M.-R. Hajirezaei, C. Klukas, H. Rolletschek, and F. Schreiber, "Integrating data from biological experiments into metabolic networks with the dbi information system." *In Silico Biol.*, vol. 5, no. 2, pp. 93–102, 2005.
- [10] A. Kerren, A. Ebert, and J. Meyer, Eds., *Human-Centered Visualization Environments*, ser. LNCS Tutorial 4417. Springer, 2007.
- [11] I. Jusufi, Y. Dingjie, and A. Kerren, "The network lens: Interactive exploration of multivariate networks using visual filtering," in *Information Visualisation (IV), 2010 14th International Conference*, July 2010, pp. 35–42.
- [12] B. Shneiderman and A. Aris, "Network visualization by semantic substrates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, pp. 733–740, 2006.
- [13] M. Wattenberg, "Visual exploration of multivariate graphs," in *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*. New York, NY, USA: ACM, 2006, pp. 811–819.
- [14] A. J. Pretorius and J. J. van Wijk, "Visual inspection of multivariate graphs," *Comput. Graph. Forum*, vol. 27, no. 3, pp. 967–974, 2008.
- [15] D. Archambault, T. Munzner, and D. Auber, "Tuggraph: Path-preserving hierarchies for browsing proximity and paths in graphs," in *Visualization Symposium, 2009. PacificVis '09. IEEE Pacific*, april 2009, pp. 113–120.
- [16] —, "Grouseflocks: Steerable exploration of graph hierarchy space," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, no. 4, pp. 900–913, july-aug. 2008.
- [17] A. Bezerianos, F. Chevalier, P. Dragicevic, N. Elmqvist, and J.-D. Fekete, "Graphdice: A system for exploring multivariate social networks," *Computer Graphics Forum (Proc. EuroVis 2010)*, vol. 29, no. 3, pp. 863–872, 2010.
- [18] Y. Wu and M. Takatsuka, "Visualizing multivariate network on the surface of a sphere," in *Asia Pacific Symposium on Information Visualisation*, 2006, pp. 77–83.
- [19] Z. Shen and K.-L. Ma, "Mobivis: A visualization system for exploring mobile data," in *Proceedings of IEEE Pacific Visualization Symposium*. IEEE VGTC, 2008, pp. 175–182.
- [20] M. Rohrschneider, A. Ullrich, A. Kerren, P. F. Stadler, and G. Scheuermann, "Visual network analysis of dynamic metabolic pathways," in *Proceedings of the 6th international conference on Advances in visual computing - Volume Part I*, ser. ISVC'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 316–327. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1939921.1939958>
- [21] C. Viau, M. J. McGuffin, Y. Chiricota, and I. Jurisica, "The flowvizmenu and parallel scatterplot matrix: Hybrid multidimensional visualizations for network exploration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, pp. 1100–1108, 2010.
- [22] Jigsaw, "Visual Analytics for Exploring and Understanding Document Collections," last accessed: 2013-02-15, <http://www.cc.gatech.edu/gvu/ii/jigsaw/datafiles.html>.
- [23] P. Eades, "A heuristic for graph drawing," *Congressus Numerantium*, vol. 42, pp. 149–160, 1984.
- [24] B. Mirkin, *Clustering for Data Mining: A Data Recovery Approach*. Boca Raton, FL, USA: Chapman & Hall/CRC, 2005.
- [25] C. A. Brewer, "Colorbrewer," <http://colorbrewer2.org/>, last accessed: 2013-02-15.
- [26] Processing.org, "Processing," last accessed: 2013-02-15, <http://processing.org/>.
- [27] MDSJ, "Java Library for Multidimensional Scaling (Version 0.2)," University of Konstanz, last accessed: 2013-02-15, <http://www.inf.uni-konstanz.de/algo/software/mdsj/>.
- [28] Trickl, "trickl-cluster," last accessed: 2013-02-15, <https://github.com/trickl/trickl-cluster>.
- [29] gicentreUtils, "giCentre Utilities," last accessed: 2013-02-15, <http://gicentre.org/utills/>.
- [30] A. Kerren and F. Schreiber, "Toward the role of interaction in visual analytics," in *Proceedings of the Winter Simulation Conference*, ser. WSC '12. Winter Simulation Conference, 2012, pp. 420:1–420:13. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2429759.2430303>