Casual Visual Exploration of Large Bipartite Graphs Using Hierarchical Aggregation and Filtering

Daniel Steinböck
TU Wien
Vienna, Austria
e0826088@student.tuwien.ac.at

Eduard Gröller
TU Wien
VRVis
Vienna, Austria
groeller@cg.tuwien.ac.at

Manuela Waldner
TU Wien
Vienna, Austria
waldner@cg.tuwien.ac.at

***Abstract*—Bipartite graphs are typically visualized using linked lists or matrices. However, these classic visualization techniques do not scale well with the number of nodes. Biclustering has been used to aggregate edges, but not to create linked lists with thousands of nodes. In this paper, we present a new casual exploration interface for large, weighted bipartite graphs, which allows for multi-scale exploration through hierarchical aggregation of nodes and edges using biclustering in linked lists. We demonstrate the usefulness of the technique using two data sets: a database of media advertising expenses of public authorities and author-keyword co-occurrences from the IEEE Visualization Publication collection. Through an insight-based study with lay users, we show that the biclustering interface leads to longer exploration times, more insights, and more unexpected findings than a baseline interface using only filtering. However, users also perceive the biclustering interface as more complex.**

***Index Terms*—information visualization, bipartite graphs, biclustering, insight-based evaluation**

I. Introduction

A bipartite graph is a special class of graphs, where the vertex (or node) set V of the graph $G = (V, E)$ can be partitioned into two disjoint nonempty sets V_1 and V_2, both of which are independent [1]. In a weighted bipartite graph, every edge connecting a node of V_1 with a node of V_2 has a weight of $\omega \geq 0$. Data sets representing bipartite graphs can be found in many disciplines, ranging from biology, where nodes represent genes and conditions [2]–[4], over document analysis, where nodes can represent different categories of named entities [5], [6], to social network analysis, where nodes can be institutions and projects [7].

Typically, visualizations of bipartite graphs can only show a few hundred nodes and edges. However, many data sets, such as the IEEE Visualization Publication collection [8], rather have thousands or ten thousands of elements. Often, these data sets are of interest to a general lay audience, such as the *Media Transparency Database*, containing all media advertising expenses of public authorities in Austria [9]. The goal of this work is therefore to find an easily understandable interactive visualization, which allows lay users to casually explore large bipartite graphs.

Common strategies to visualize large graphs – and large data sets in general – are *filtering* (i.e., removing items) and *aggregation* (i.e., grouping items) [10], [11]. In this paper, we propose a new interactive visualization technique (BiCFlows) combining aggregation and filtering particularly for large bipartite graphs. We use hierarchical aggregation, where elements are iteratively aggregated into groups, and the user can gradually drill down from an overview to the finest detail level [11]. Since we work with bipartite graphs without inherent hierarchical groupings, we use biclustering to partition the graph into topologically meaningful groups. Depending on the exploration level and group size, we filter the groups to show only the most relevant items. We explore the usefulness of BiCFlows by showcasing two application cases and let lay users explore a publicly available data set in an insight-based evaluation.

In summary, our paper has two main contributions:

1) a new visualization and interaction design for casual exploration of large, weighted bipartite graphs and
2) the results of an insight-based user study, where lay users explored a large bipartite graph containing advertising expenses of public organizations.

II. Related Work

The most common visual encodings of graphs are node-link diagrams and matrix-based representations [10], [12]. For bipartite or k-partite graphs, nodes of the k different sets can be differentiated by color in node-link diagrams (see, for instance, the graph view in *Jigsaw* [5]), or nodes of one set can be attracted to nodes of the other set, anchored at fixed locations [13], [14]. Bipartite graphs shown as biadjacency matrices have rows and column keys corresponding to the nodes of the two independent sets, while cells represent their connections (e.g., the scatterplot view in *Jigsaw* [5] or the network matrix by Dormann et al. [15]). Another common way

This work was financed by the Austrian Science Fund (FWF): T 752-N30. This paper was partly written in collaboration with the VRVis Competence Center. VRVis is funded by BMVIT, BMWFW, Styria, SFG and Vienna Business Agency in the scope of COMET - Competence Centers for Excellent Technologies (854174) which is managed by FFG.

to visualize k-partite graphs are adjacent lists, where the nodes of the independent sets represent the list entries and edges between these lists show their connections [5], [7], [15]–[17]. While this visual encoding is easily understandable and allows for efficient scanning of the node labels, it does not scale well with the number of nodes. To deal with a larger number of nodes than fit to the screen, these examples use scrolling [5], filtering according to node attributes [17], or focus+context representations [16]. However, if there are thousands of nodes in one set, these approaches lead to extensive interaction effort, information loss, or visual clutter.

To overcome these limitations, hierarchical aggregation techniques for large graphs have been proposed. For instance, *GrouseFlocks* iteratively constructs a graph hierarchy through attributes of the underlying graph data [18]. The user can then interactively create cuts through the graph hierarchy and visualize the cut graph in a node-link diagram with aggregated meta-nodes. Alternatively, aggregated meta-nodes can be visualized as matrices embedded within a node-link diagram [19], [20] or as zoomable adjacency matrices [21]. These examples aggregate nodes either based on node attributes or based on topological properties, such as graph cliques.

For our system, we assume that we do not have any additional node attributes that can be used for constructing a graph hierarchy. We therefore use *biclustering* [22] (or *co-clustering* [23]), which finds groups of coherent items in bipartite graphs. Biclustering is mostly used in bioinformatics for studying gene expression data [2]–[4] and in document classification [23], [24]. Essentially, biclustering simultaneously rearranges rows and columns of the biadjacency matrix to form clusters of certain similarity. Visualization of biclustered graphs often use color-coded matrices [25]–[27], node-link diagrams with cluster enclosings [28], or matrices embedded into node-link diagrams [29]–[31]. Biclustering has also been used to bundle edges of bipartite graphs shown in adjacent lists [6], [32]. Edge bundling of adjacent lists can improve the perception of the visualization and the quality of the analysis [33]. Since only edges are bundled, these lists still do not scale well with the number of nodes.

For better scalability in terms of nodes, Zhao et al. [34] recently introduced *BiDots*, which also uses adjacent lists, but groups nodes row-wise based on their associated biclusters. Nodes are represented as circles with unique line patterns, which can also co-occur in multiple clusters. Using this compact representation, they successfully visualized named entities extracted from a text corpus with a few hundred entities per set and around 1,000 connections between these sets. However, it is unclear how well the visualization scales to a data set with thousands of nodes per set and ten thousands of connections as in our use cases.

VIBR [35] is able to visualize bipartite graphs with such a size using an adjacency list of node clusters, which can be explored hierarchically. In contrast to BiCFlows, this visualization requires a legend for mapping colors to node labels. We chose to adopt the more common list-based representation, which supports direct labeling of the cluster blocks and therefore reveals more information on the first glance.

Another recent approach to visualize very large unweighted bipartite graphs by Pezzotti et al. [36] also uses hierarchical aggregation. They introduce a novel adaptation of the hierarchical stochastic neighbor embedding (HSNE) algorithm, and place landmark vertices of HSNE clusters in two parallel axes connected by edges. These landmarks can be brushed to reveal lower hierarchy levels. With their C++ implementation, Pezzotti et al. visualize bipartite graphs with millions of nodes. Since our focus lies on casual exploration, we do not extract landmarks to represent clusters, but use a filtering approach within each cluster to show the most relevant elements in terms of weights per cluster. This way, we can also show node labels so that the user can see some relevant information on a single glance, without having to interact.

III. Visualization and Interaction Design

The main requirements for the visualization and interaction design of BiCFlows were:

1) the visualization should scale up to thousands of nodes and edges,
2) it should support in-depth exploration of the data, such as identifying clusters of similar elements of varying size and retrieving connections of a selected element,
3) it should provide some initial information on the first glance, and
4) the visualization and interaction design should be easily understandable for a lay audience.

To achieve Requirements 1 and 2, we use a combination of hierarchical aggregation and filtering. Aggregation is achieved using hierarchical biclustering, while filtering is performed based on ranking of accumulated edge weights (Section III-A).

To fulfill Requirements 3 and 4, we opted for a visualization using adjacent lists (Section III-B). Lists are ubiquitous and therefore presumably easy to understand for visualization novices. In addition, they can feature sufficiently large text labels so that users can gain some initial understanding on the first glance. In contrast, the similarly popular matrix view of bipartite graphs requires very short or 90° rotated column labels. We describe the interaction design to interactively drill down into the biclustering hierarchy and obtain details-on-demand in Section III-C.

A. Hierarchical Biclustering

A bipartite graph can be viewed as a weighted biadjacency matrix, where rows represent nodes of set V_1, and columns represent nodes of the other set V_2. Each matrix cell contains the corresponding edge weight between two nodes from V_1 and V_2. Biclustering rearranges the rows and columns of the matrix to create coherent blocks (see Figure 1). Biclustering is an NP-hard problem [37], but many algorithms that optimize search heuristics have been developed. In our system, we use an algorithm, which tries to maximize the modularity of the bipartite graph for a predefined number of clusters [38]. Modularity describes how densely nodes are connected in a partition compared to the rest of the graph. In contrast to

other biclustering algorithms, this approach can also handle weighted biadjacency matrices.

Biclustering algorithms assume a specific structure of the underlying data matrix. Commonly used structures are the *block diagonal structure*, where each row and column is assigned to exactly one cluster, and the *checkerboard structure*, where each row and column is assigned to multiple clusters, so that each cell is assigned to exactly one cluster. For our system, we use a block diagonal structure so that each node is associated with only a single cluster (see diagonal blocks in Figure 1).

As we are dealing with large bipartite graphs, we filter the number of visualized nodes and their associated edges within a cluster based on their cumulated edge weights. If a user wants to reveal filtered nodes and edges, she can select the desired cluster to further drill down into the data. The system then biclusters the sub-matrix of the selected cluster and subsequently visualizes those items, as well as nodes from other clusters connected to at least one node from the selected cluster (Figure 1). The higher the modularity of the clusters, the fewer nodes have edges across cluster boundaries. If a cluster has no edge to another cluster, it represents a disconnected subgraph of the original graph.

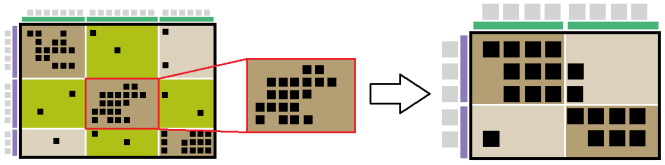

Fig. 1. A biadjacency matrix with three clusters shown in dark brown (left) and one cluster selected (red). The selected sub-matrix is further biclustered (right). The four lime-green sub-matrices (left) contain edges between nodes of the selected cluster and nodes of other clusters.

By subsequently selecting sub-clusters, the users can interactively drill down from the initial overview to a subset of the data, which cannot be further subdivided into smaller clusters. This is the case if the biadjacency matrix to be clustered has only a single row or column, or if the bipartite graph is too dense to be clustered further.

B. Visual Encoding

To visualize bipartite graphs, we use two parallel, vertical lists of nodes, where – similarly as for Sankey diagrams [39] and parallel sets [40] – the thickness of an edge connecting two nodes is defined by its edge weight, and the rank of each node is defined by its accumulated edge weights $\sum \omega_i$ (Figure 2). Nodes are grouped according to biclusters and ordered according to their accumulated edge weights within each cluster and list, respectively. Since we initially display a large number of nodes per cluster, we filter nodes with small edge weights. Given the sum of all edge weights in the entire graph $\sum \omega$, the smallest displayable unit for a node h, and

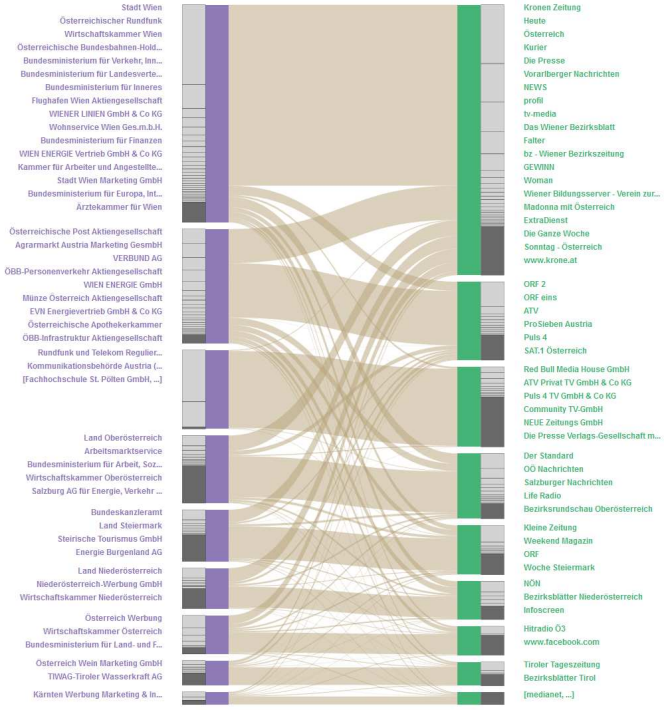

Fig. 2. BiCFlows showing individual nodes (gray), aggregated filtered nodes (dark gray), clusters of nodes (purple and green), as well as their connections.

the total height H of the visualization, we only display nodes, which fulfill the following criterion:

$$\sum \omega_i \geq \frac{h \sum \omega}{H}. \tag{1}$$

In other words, we filter all nodes that would be encoded smaller than the smallest acceptable height in the list h. For our use cases, we set h to two pixels and adapted the height of the visualization H dynamically to the display size.

Since text labels are important to get a quick initial overview of the data, we try to maintain as many node labels as possible. For each group of clustered nodes, we therefore vertically stack as many node labels as possible next to the cluster group (indicated by purple and green bars in Figure 2). Since we use labels with a fixed font size but variable node heights, the number of labels may differ from the number of visualized nodes, and labels may be shifted from their associated nodes. To obtain more details about a labeled node, users can hover the label or the node itself to reveal all its connections (Figure 3).

C. Interactive Exploration

Our system supports two basic exploration mechanisms: *highlighting* and *drill-down*. Users can request details-on-demand by either hovering nodes and node labels, respectively, or cluster bars. In the first case, all connections of one individual node are highlighted (see Figure 3). In the second case, only connections of nodes in the hovered cluster are visualized. In the example of Figure 4, we can see that the

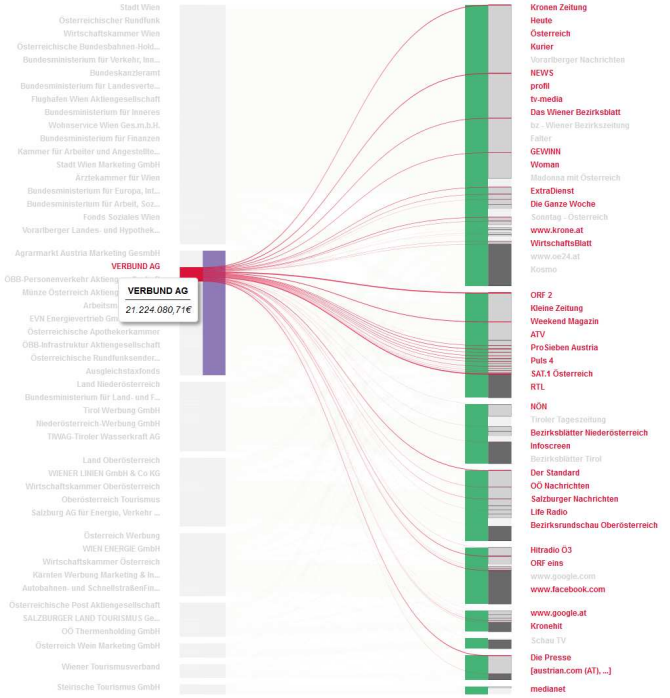

Fig. 3. When hovering a node label (left), all edges of the node are visualized, and visible labels of the connected nodes (right) are highlighted in red.

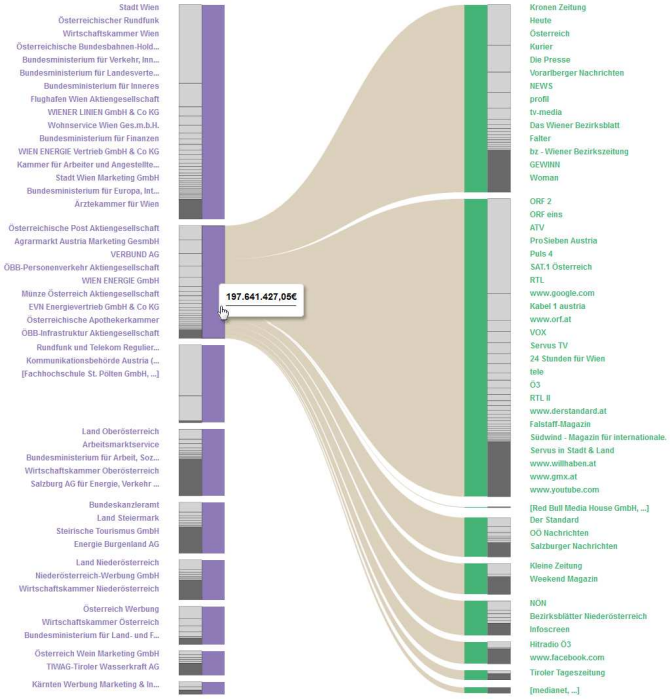

Fig. 4. Hovering a cluster reveals only those nodes in the adjacent list, which have connections to the hovered cluster.

hovered cluster has strong connections to multiple clusters, and therefore a rather low modularity and many edge crossings, respectively.

Drill-down is necessary to reveal filtered nodes from selected clusters through iterative biclustering as described in Section III-A. When selecting a cluster, all nodes that do not have connections to the nodes in the selected cluster are discarded (light brown sub-matrices in Figure 1). Nodes of other clusters connected to nodes in the selected cluster (lime-green sub-matrices in Figure 1) are aggregated into one group (lime-green group on the bottom in Figure 5). To help users maintain orientation and keep an overview, we provide context bars on the side (Figure 5), where each bar shows the selected cluster among the grayed out unselected clusters. These bars are also used to navigate back to a higher hierarchy level. In Figure 5, the user selected the small bottom cluster in the initial overview (indicated by the small purple and green bars on the outermost context bars).

If the graph to be visualized is dense, it may not be possible to further sub-cluster the data without filtering nodes, as the biadjacency sub-matrix is already complete. This means that some nodes with small accumulated edge weights will never be visualized. For this case, we additionally provide a ranked text list of node labels, which can also be searched, to browse all nodes, including filtered ones. Selecting an invisible node highlights its connections in the visualization as shown in Figure 3.

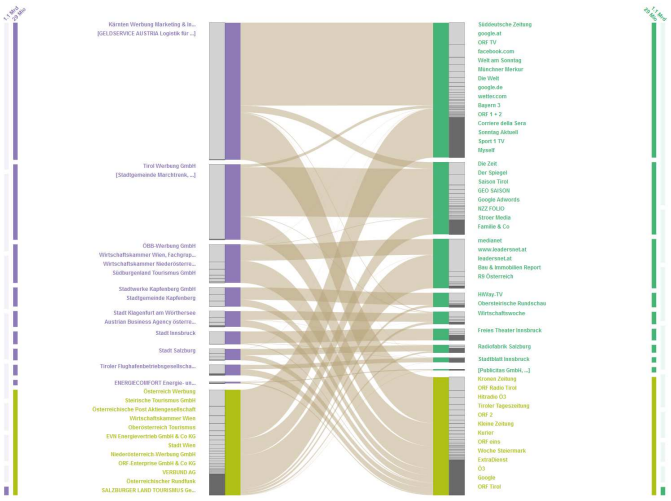

Fig. 5. Selected cluster divided into nine subclusters, with connections to other clusters (lime-green) and the context bars on the side providing an overview of the entire data set.

D. Implementation

BiCFlows is implemented using a client-server infrastructure to separate the computationally expensive biclustering from the user interface on the client side. The server is implemented with *Python* and *Numpy* for efficient processing of large data structures. We use the Python implementation of *CoClust* [38] to cluster the biadjacency matrices. As our system is intended for visual exploration by lay users, we host individual data sets as separate web services. For each data set,

we use CoClust to determine the optimal number of clusters in a precomputation step. In this step, CoClust computes multiple clusters and finds the resulting biadjacency matrix with the maximum modularity. For small browser windows, we decrease the number of clusters to avoid visual clutter. The actual biclustering is performed online, and sub-clustering is invoked whenever a user selects a cluster. Depending on the number of nodes in the graph, biclustering can take up to several seconds. On a consumer hardware (Intel i7 CPU with 4 GHz and 8 GB RAM), biclustering of a biadjacency matrix of 4,976 rows and 2,120 columns (data set described in Section IV-B) takes around two seconds. The initial biclustering of the entire data set is only performed once when loading the page. Whenever the user selects a cluster, the biclustering results of the higher hierarchy levels are locally stored, so that the user can quickly navigate back to previous views.

The client was implemented using *D3.js* [41] based on an existing bipartite layout [42]. The visualization is embedded within multiple coordinated views, which also contain a text list of nodes to select highlighted nodes and bar charts showing additional data attributes. The bar charts can be used to filter the data, e.g., according to time. *Crossfilter* [43] was employed to quickly filter the data set on the client side.

IV. Use Cases

We will showcase the usefulness and discuss potential limitations of BiCFlows using two data sets:

1) the so-called *Media Transparency Database* [9], where all public authorities of Austria have to report their advertisement expenses to media companies above 5,000 Euros beginning from 2012 (Section IV-A), and
2) the *IEEE Visualization Publication* collection [8], where meta-data of all major IEEE visualization papers since 1990 are collected. Using BiCFlows, we visualize the author-keyword relations (Section IV-B).

Both data sets have thousands of nodes and fulfill the properties of a weighted, bipartite graph. The visualizations can be accessed online [44].

A. Media Transparency Database

The Austria Media Transparency Database [9] is of great interest to journalists to reveal relations between public and media organizations, by media organizations themselves to investigate their competitors, and to the general public to find out how their tax money is spent. The database is updated quarterly, and journalists regularly parse the database for new interesting money flows. In particular, they are interested to find out if certain ministries advertise in similar media and which ministries spend a high amount of money for advertisement. However, finding this information is tedious, since names of ministries change over legislation periods, and some big media organizations comprise dozens of sub-companies, which all show up as separate entities in the database.

By 2017, the Media Transparency Database contained 1,200 legal entities, reporting advertising expenses to over 4,200 media organizations. In total, the database has more than 34,000 entries. The reported expenses are not evenly distributed, with very few very high values (e.g., almost 20 million Euros aggregated advertisement expenses issued from the government of the city of Vienna to the daily newspaper *Kronen Zeitung*), and most of the expenses around 5,000 Euros. The highest modularity (0.5) was found for nine clusters (see Figure 2).

Due to the large public interest, there are already a few online visualizations of the Media Transparency Database available, such as a dashboard visualization by Rind et al. [45] and a web service by Salhofer et al. [46]. However, these existing visualizations rely solely on filtering of the data and therefore only visualize a very small fraction of the existing entities. With these visualizations, users can get information about the most relevant legal entities and media organizations. However, smaller transactions, for instance because advertising expenses are spread across multiple smaller media organizations, are not visualized.

Like these previous approaches, BiCFlows reveals important legal entities and media organizations on the first glance. The two top-most labels in Figure 2 show the legal entity (*Stadt Wien*) and media organization (*Kronen Zeitung*) spending and receiving the highest accumulated sums, respectively. These two nodes are grouped into the same cluster with other popular Austrian newspapers, such as *Heute* or *Kurier*, and other legal entities spending high amounts for advertising in these daily newspapers. The second-ranked legal entity (*Rundfunk und Telekom Regulierungs-GmbH*) is contained in a different cluster, which is ranked third in Figure 2. Selecting the cluster reveals that this legal entity mainly sponsors small radio and TV stations, where most of them do not receive any advertisement money from other legal entities. If only the 10 top-ranked media organizations were shown, not a single media organization receiving money from this legal entity would be visualized.

When drilling into the data, a frequently occurring grouping reveals geographic proximity. Often, the groupings contain smaller legal entities and media organizations located in the same regions by just moving one hierarchy level down. This is not surprising, since smaller entities tend to advertise in smaller and more local media. Other clusters are related topic-wise. For instance, drilling down three hierarchy levels reveals a cluster of many media organizations related to air travel, such as *Airline Business* or *Air Transport World* associated with a single legal entity – the *Vienna International Airport*. We used the Media Transparency Database for our user study, and also report some of the resulting insights of the participants in Section V.

B. IEEE Visualization Publications

Co-authorship networks are a common use case for graph visualization, such as by Henry et al. [19]. Using the IEEE visualization publication collection by Isenberg et al. [8], we pursue a different approach to investigate commonalities

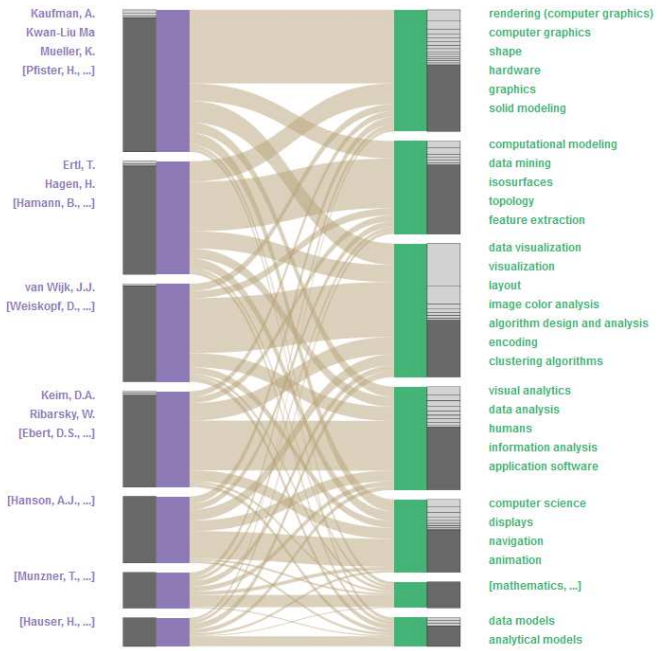

Fig. 6. Seven biclusters of authors and IEEE key terms of IEEE visualization publications (data from Isenberg et al. [8]).

between authors. We retrieved 4,976 authors from the data set, as well as 2,120 IEEE key terms these authors used to classify their papers. Biclustering is employed to reveal groups of authors that tend to use similar key terms – or, conversely, groups of key terms that tend to be used by the same authors. We group the data set into seven clusters, but the modularity of this data set is rather low (0.31 for seven clusters). In other words, the topical groups defined by the key terms are not coherent across authors.

Figure 6 shows the visualization of the seven clusters, revealing topics around rendering, modeling, visualization and layout, visual analytics, displays and navigation, mathematics and bioinformatics, and data models, as well as their associated main authors. Selecting these clusters can yield interesting sub-topics. For instance, selecting the bottom cluster in Figure 6 reveals the top sub-cluster with application-specific key terms (Figure 7 top). Sub-clustering this cluster again reveals a cluster of key terms from the automotive industry with its associated main authors (Figure 7 bottom).

This example also explains why the clusters have a rather low modularity: While K. Matkovic is the most common co-author of H. Hauser according to DBLP [47], his publication keywords are much broader than suggested by this clustering. Highlighting all key terms used by H. Hauser by hovering his name, we discover that, in fact, his most commonly used key term in the IEEE Visualization Publication data set is *data visualization* (used 19 times), followed by *computational modeling* (used 12 times). The most commonly used key term in the bottom cluster of Figure 7 (*"engines"*) was used only four times by H. Hauser. This means that BiCFlows

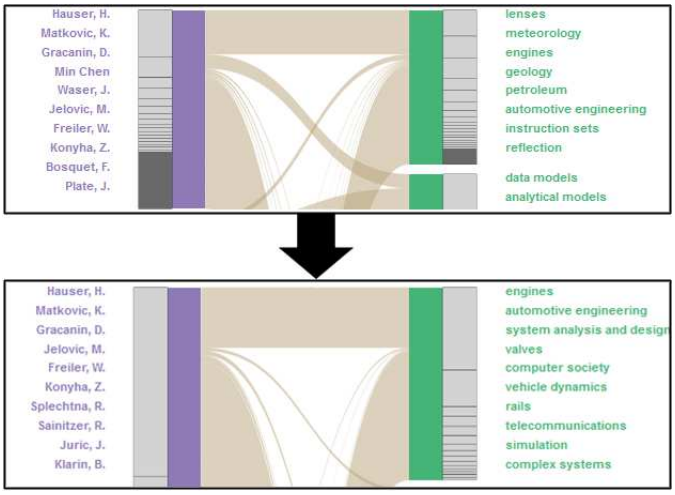

Fig. 7. Two steps of sub-clustering on the bottom cluster in Figure 6.

is able to reveal meaningful clusters of key terms in this example. However, the clusters themselves are not necessarily representative for individual nodes.

An alternative biclustering approach, such as biclustering based on a checkerboard structure (see Section III-A), could solve this limitation by assigning graph nodes to multiple biclusters. In this case, the number of nodes to display would linearly increase with the number of clusters.

V. User Study

We conducted an insight-based evaluation [48] to formally compare the benefits and limitations of BiCFlows for visualizing large bipartite graphs with a simple filtering approach. We used the Media Transparency Database introduced in Section IV-A for our evaluation. We recruited 12 users (four females, eight males, aged 25 to 56) with different backgrounds, including one computer scientist, and all experienced computer and internet users. One user had prior knowledge of the Media Transparency Database, two had heard of it before, and nine did not know it at all. However, all users were roughly familiar with the political and media landscape in Austria.

As a baseline condition, we used a simplified version of BiCFlows, which reduces the number of displayed items solely by filtering, but does not perform any aggregation (see Figure 8). Here, we refer to this baseline as *Cut-Off*. All nodes that are too small to be labeled are aggregated into a single "others" node (the bottom nodes in Figure 8). The Cut-Off visualization allows for highlighting of selected nodes like BiCFlows. Legal entities and media organizations that are filtered can be selected from the linked text list to visualize all associated advertisement expenses.

A. Hypotheses

Our main goal has been to investigate the benefits and limitations of the combined aggregation and filtering approach of BiCFlows compared to a simple filtering approach, which is the common method to visualize the Media Transparency

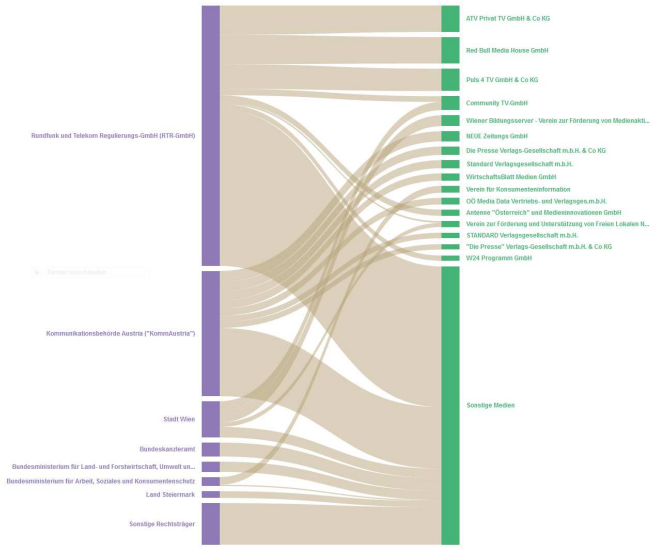

Fig. 8. The baseline condition of the study (Cut-Off) using only filtering, but no aggregation.

Database [45], [46]. Our assumption has been that iteratively drilling down into the aggregated data would encourage lay users to casually explore the visualized data in more detail and, as a consequence, gain more knowledge. On the other hand, we also assumed that BiCFlows would be perceived as more complex and harder to use than the Cut-Off baseline visualization. We therefore formulated two main hypotheses:

H1: *With BiCFlows, users will gain more insights than with Cut-Off.*

In particular, we expected that users would discover more legal entities and media organizations, as well as transactions between them (**H1.1**), that they would mention more entities with small accumulated advertisement sums (**H1.2**), establish more connections between entities or reason about commonalities (**H1.3**), discover more unknown entities or unexpected information (**H1.4**), and spend more time exploring the data (**H1.5**).

H2: *BiCFlows will be perceived as more complex than Cut-Off.*

B. Design

We employed a within-subjects design with visualization as independent variable, with the two levels BiCFlows (BiC) and Cut-Off (CO). The presentation order of the two visualizations was counter-balanced.

We used two subsets of the Media Transparency Database for the evaluation. The first data set, comprising only advertising objectives, contained 1,226 legal entities and 3,544 media organizations. We used nine clusters with a modularity of 0.39. The second data set, containing only press subsidies, had 68 legal entities and 885 media organizations. We also used nine clusters, yielding a modularity of 0.62. This means that the second data set was smaller with more coherent groups. The assignment of the two data sets to the two visualizations was also counter-balanced.

TABLE I
CODING CATEGORIES OF THINK-ALOUD PROTOCOLS.

Code	Description
Entities	A mentioned legal entity or media organization.
Sums	Mentioned transaction sums between one legal entity and one media organization, or a total sum spent by a legal entity or received by a media organization.
Duplicates	Discovered entities with same or similar name, e.g., *google.at* and *google.de*, where users explicitly mentioned that these are the same.
Time	Quarters, years, or periods mentioned.
Geography	Geographical connections made for certain entities, e.g., "*DORF TV* is probably from Upper Austria too, because it's in the same group as other media organizations from Upper Austria."
Comparisons	Comparisons between entities or time periods, e.g., "*ÖBB* spent 19 million Euros, but compared to *Stadt Wien* that's nothing."
Reasoning	Generating hypotheses to explain an observation, e.g., "*Heute*, *Krone*, and *Österreich* receive most money, that's probably because they have most readers."
Unknown Entities	Entities that were unknown to the user, e.g., "*a3ECO*? - Never heard of it before."
Unexpected Findings	Unexpected findings or astonishments, e.g., "I can't believe *Stadt Wien* spends that much money."

The study was conducted using the Mozilla Firefox web browser on a 27" monitor. Users had to fill out a consent form, a demographic questionnaire, and then read a printed task description. Every condition was preceded by a training period using a test data set. At the end of the evaluation, users had to fill out a post-experiment questionnaire.

C. Analysis

Insight has been defined as *"individual observation about the data by the participant"* [49]. To reveal whether users made observations, insight-based evaluations use an open-ended think-aloud protocol, which afterwards is coded and quantified for formal evaluation [48]. Users are encouraged to explore the data as long as they think they can find something new.

We recorded all user sessions using screen capturing and audio recording, and encouraged the participants to comment on everything they see or experience during the data exploration. After the experiment, we transcribed the audio recordings and performed open coding, yielding nine insight categories listed in Table I. For each user, we aggregated the number of codes per condition and used these numbers for comparing insights to verify hypotheses H1.1-H1.4.

In addition, we also recorded the exploration time (H1.5) and the users' subjective usability ratings through the *System Usability Scale* (SUS) questionnaire [50] (H2). All obtained measures were statistically analyzed using Wilcoxon Signed-Rank tests.

D. Results

To test hypothesis H1.1, we compared the number of *unique* entities mentioned by the users. Using BiC, users mentioned significantly more different entities compared to CO ($Z = 10.5, p = .045$, Figure 9(a)). They also mentioned significantly more transaction sums ($Z = 1.5, p = .005$, Figure 9(b)). *We*

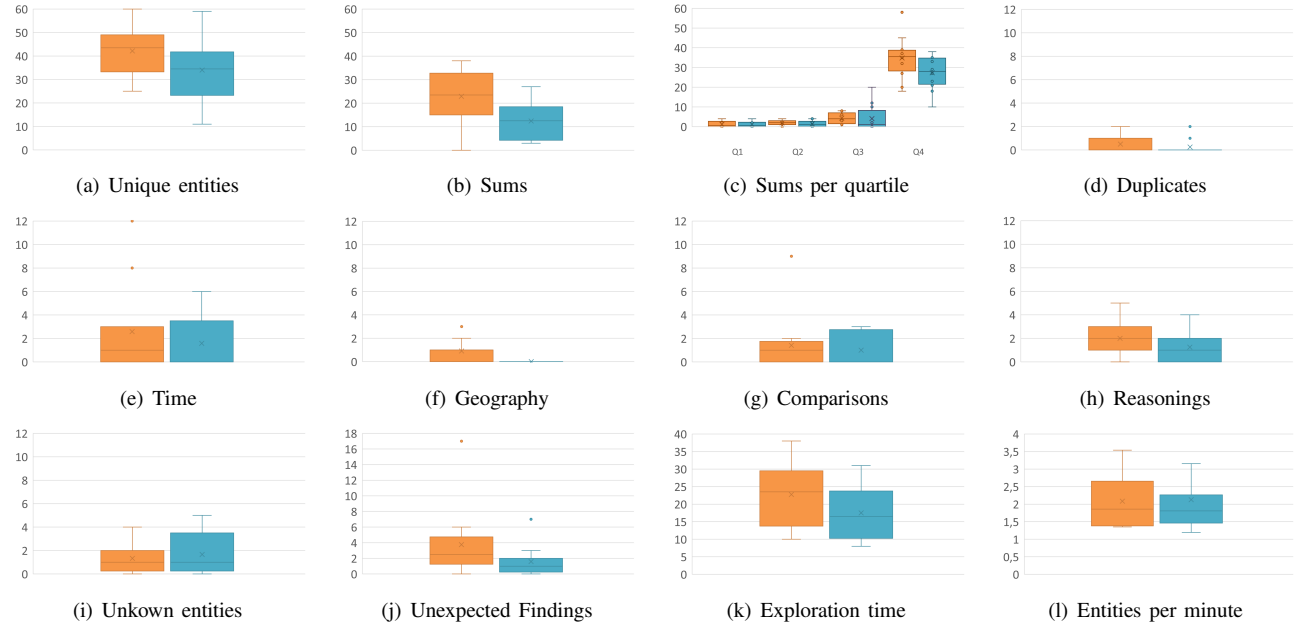

Fig. 9. Box plots of the number of coded insights per category (Table I), as well as exploration times in minutes (k) and mentioned unique entities per minute (l). The left orange box plot shows the results of BicFlows, the right blue one of the Cut-Off visualization.

can thereby confirm our hypothesis H1.1: Users mention more entities and transaction sums using BiCFlows.

For hypothesis H1.2, we calculated the quartiles of all cumulated entity sums and compared the number of mentioned entities separately for the lower three quartiles. However, the number of mentions is almost equivalent for *Q1-Q3* (see Figure 9(c)). *This disproves our hypothesis H1.2: Users do not mention more entities with smaller transaction sums using BiCFlows.*

To verify H1.3, we looked at utterances coded as duplicates, time, geography, comparisons, and reasonings. Detection of duplicates was generally low, and the difference between the two conditions is not significant ($Z = 2, p = .257$, Figure 9(d)). Mentions of temporal relations were a little bit more common, but also comparable between the conditions ($Z = 14, p = .310$, Figure 9(e)). While no user made any remark on geographic connections using CO, there were a few mentions of geographic relations using BiC (see Figure 9(f)). Finally, users did not make significantly more comparisons in either interface ($Z = 16, p = .774$, Figure 9(g)) and did not significantly reason more about the data using BiC ($Z = 20, p = .234$, Figure 9(h)). *We can therefore partially confirm H1.3: users discovered some geographic connections between entities using BiC and none with CO, but they did not find significantly more duplicates, temporal relations, or other commonalities or differences between entities.*

For hypothesis H1.4, we compared the number of unknown entities and unexpected findings. There is no significant difference between the number of unknown entities discovered in the data set ($Z = 18.5, p = .633$, Figure 9(i)). However, users discovered more unexpected information, which was indicated by astonished or disbelieving reactions, using BiC than using CO ($Z = 8, p = .045$, Figure 9(j)). *Thus, we can partially confirm our hypothesis H1.4: Users discovered more unexpected information using BiCFlows, but did not find more unknown entities.*

To test H1.5, we compared the time each user spent exploring the two different interfaces. Users spent more time exploring data using BiC (23 minutes on average) than CO (17.5 minutes), which is a significant difference ($Z = 6.5, p = .032$, Figure 9(k)). The average number of unique entities mentioned per minute, however, is very similar ($Z = 35, p = .754$, Figure 9(l)). *This confirms hypothesis H1.5: Users did not discover entities at a faster rate using BiC, but rather spent a longer time exploring the data.*

Finally, we compared the users' ratings of the SUS questionnaire to test hypothesis H2. With an average SUS score of 82, CO was rated significantly higher than BiC with 72 ($Z = 4, p = .028$). *This confirms our hypothesis H2: Users perceived BiCFlows as more complex than the Cut-Off approach.*

E. Discussion

In summary, our study showed that users explored the visualization for a longer time using BiCFlows than the Cut-Off visualization, which does not use any hierarchical aggregation. The rate of insights per minute was comparable. This means that users discovered more entities (i.e., nodes) and more transaction sums (i.e., edges) when exploring the Media Transparency Database using BiCFlows because they were encouraged to perform longer explorations. In particular, they made more unexpected findings.

This higher number of insights, however, comes with a lower perceived usability. While both interfaces were rated as excellent according to SUS [50], the issued average scores are on the upper and lower bounds of the excellent rating, respectively. Informal feedback indicates that users found the clustering irritating at the beginning, but gained sufficient understanding after exploring for a while.

Surprisingly, users did not find more entities with smaller expenses using BiCFlows than the baseline. We initially assumed a major strength of hierarchical aggregation would be that the user can reveal those lower ranked nodes and edges by drilling down. In contrast, in the Cut-Off visualization, these nodes never show up. From the video recordings, one observation was that participants usually only drilled down one or two hierarchy levels. Entities with low accumulated edge weights are potentially not yet revealed. Using BiCFlows, most users did not interact with the text lists and the linked bar charts at all. In contrast, the major exploration interface of the Cut-Off approach was not the visualization itself, but the text list of ranked legal entities and media organizations. When using the Cut-Off approach, most users scrolled these lists far down and mentioned entities from these lists while scrolling.

The most common unexpected findings across both conditions were the advertising expenditures of the daily newspapers *Kronen Zeitung*, *Heute*, and *Österreich*, as well as irritation about the fact that the more popular TV station *ORF1* receives less money than the smaller *ORF2*. However, users of BiCFlows mentioned more often that *Stadt Wien* advertises in a large number of media. We assume it is due to the aggregation into the large "others" group in the Cut-Off visualization that users cannot easily grasp the true number of edges of a selected node.

F. Limitations

One limitation of our study is the potentially confounding factor introduced by the different text label strategies of BicFlows and Cut-Off. While we try to maximize the number of node labels per group in BiCFlows, we assign a single node label to each node in the Cut-Off visualization (see Figure 8). This resulted in up to three times as many node labels in BiCFlows compared to the Cut-Off visualization. This can be an alternative explanation for the higher number of mentioned entities using BiCFlows.

Since we did not systematically vary the data characteristics, our study also does not reveal how the size of the data set and the modularity of the clusters influence the effectiveness and understandability of the visualization. With more data, the system response will be slower and users will have to perform more interaction steps to reveal lower ranked nodes. With lower modularity, the meaningfulness of the visualized clusters will decrease and may lead to misinterpretations of the data.

Generally, we did not thoroughly evaluate the quality of the coded comparisons, reasonings, and temporal or geographical insights. In the future, it will be interesting to encourage users to characterize the commonalities of cluster elements to assess whether they correctly interpret the grouping. An example would be whether users believe that clusters were derived based on geographical locations of entities and incorrectly conclude that all legal entities and media organizations of a certain region are present in a selected cluster.

VI. Conclusions

We presented a novel approach for visualizing large bipartite graphs by combining hierarchical aggregation through biclustering and filtering in adjacent lists. With two use case examples, we demonstrated how BiCFlows supports interactive exploration of bipartite graphs with thousands of nodes and edges. From our evaluation, we conclude that the major strength of BiCFlows is the encouragement of users to perform a deeper exploration of the data. As a consequence, they have more insights and discover more unexpected information. The limitation of BiCFlows is a higher cognitive demand – at least initially – and a lower perceived usability for a lay audience. We also observed that users generally only drill down one or two hierarchy levels.

Based on these observations, we conclude that hierarchical aggregation is beneficial if the goal is to encourage users to perform a deep exploration of a large bipartite graph to discover unexpected information. However, if the goal is to provide a simple interface to primarily look for specific entities, a visualization based solely on filtering combined with a search tool seems to be the more promising option.

The usefulness of BiCFlows furthermore depends on two factors: the size of the data set and the modularity of the clustering. The bipartite graphs in our use cases had thousands of nodes and edges. For bipartite graphs with millions of nodes, the initial biclustering of the entire data set will take longer than the tolerable waiting time of a few seconds for a web application. For larger data sets, it will therefore be necessary to use a faster clustering method (e.g., the adopted HSNE algorithm by Pezzotti et al. [36]) or to precompute the clusters. In addition, users will have to drill down more hierarchy levels to reveal nodes with low cumulated edge weights. If the modularity is low because there is no clear topological grouping inherent to the data, the visualization has a lot of edge crossings and a lot of connections to other sub-clusters (lime-green bars in Figure 5). In the future, we therefore plan to evaluate alternative clustering methods for different graphs and use cases.

References

[1] M. S. Rahman, *Basic Graph Theory*, ser. Undergraduate Topics in Computer Science. Cham: Springer International Publishing, 2017.

[2] S. C. Madeira and A. L. Oliveira, "Biclustering algorithms for biological data analysis: a survey," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 1, no. 1, pp. 24–45, Jan. 2004.

[3] D. Jiang, C. Tang, and A. Zhang, "Cluster analysis for gene expression data: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 11, pp. 1370–1386, Nov. 2004.

[4] B. Pontes, R. Giráldez, and J. S. Aguilar-Ruiz, "Biclustering on expression data: A review," *Journal of Biomedical Informatics*, vol. 57, pp. 163–180, Oct. 2015.

[5] J. Stasko, C. Görg, and Z. Liu, "Jigsaw: Supporting Investigative Analysis through Interactive Visualization," in *2007 IEEE Symposium on Visual Analytics Science and Technology*, Oct. 2007, pp. 131–138.

[6] M. Sun, P. Mi, C. North, and N. Ramakrishnan, "BiSet: Semantic Edge Bundling with Biclusters for Sensemaking," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 310–319, Jan. 2016.

[7] S. Ghani, B. C. Kwon, S. Lee, J. S. Yi, and N. Elmqvist, "Visual Analytics for Multimodal Social Network Analysis: A Design Study with Social Scientists," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2032–2041, Dec. 2013.

[8] P. Isenberg, F. Heimerl, S. Koch, T. Isenberg, P. Xu, C. D. Stolper, M. M. Sedlmair, J. Chen, T. Möller, and J. Stasko, "vispubdata.org: A Metadata Collection about IEEE Visualization (VIS) Publications," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 9, pp. 2199–2206, 2017.

[9] Bundeskanzleramt, "BGBl. I Nr. 125/2011," https://www.ris.bka.gv.at/eli/bgbl/I/2011/125/20111227, 2011, [Online; accessed Sep-2018].

[10] T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. J. van Wijk, J.-D. Fekete, and D. W. Fellner, "Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges," *Computer Graphics Forum*, vol. 30, no. 6, pp. 1719–1749, Sep. 2011.

[11] N. Elmqvist and J.-D. Fekete, "Hierarchical Aggregation for Information Visualization: Overview, Techniques, and Design Guidelines," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 3, pp. 439–454, May 2010.

[12] I. Herman, G. Melançon, and M. S. Marshall, "Graph visualization and navigation in information visualization: A survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, no. 1, pp. 24–43, Jan. 2000.

[13] K. Misue, "Drawing Bipartite Graphs As Anchored Maps," in *Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation - Volume 60*. Australian Computer Society, Inc., 2006, pp. 169–177.

[14] M. Dumas, M. J. McGuffin, J.-M. Robert, and M.-C. Willig, "Optimizing a Radial Layout of Bipartite Graphs for a Tool Visualizing Security Alerts," in *International Symposium on Graph Drawing*. Springer, Berlin, Heidelberg, 2011, pp. 203–214.

[15] C. F. Dormann, J. Fründ, N. Blüthgen, and B. Gruber, "Indices, graphs and null models: analyzing bipartite ecological networks," *The Open Ecology Journal*, vol. 2, no. 1, 2009.

[16] H.-J. Schulz, M. John, A. Unger, and H. Schumann, "Visual Analysis of Bipartite Biological Networks," in *Proceedings of the First Eurographics Conference on Visual Computing for Biomedicine*. Eurographics Association, 2008, pp. 135–142.

[17] M. Dörk, N. H. Riche, G. Ramos, and S. Dumais, "PivotPaths: Strolling through Faceted Information Spaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2709–2718, Dec. 2012.

[18] D. Archambault, T. Munzner, and D. Auber, "GrouseFlocks: Steerable Exploration of Graph Hierarchy Space," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 4, pp. 900–913, Jul. 2008.

[19] 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, no. 6, pp. 1302–1309, Nov. 2007.

[20] S. Rufiange, M. J. McGuffin, and C. P. Fuhrman, "TreeMatrix: A Hybrid Visualization of Compound Graphs," *Computer Graphics Forum*, vol. 31, no. 1, pp. 89–101, Feb. 2012.

[21] N. Elmqvist, T.-N. Do, H. Goodell, N. Henry, and J.-D. Fekete, "Zame: Interactive large-scale graph visualization," in *IEEE Pacific Visualization Symposium, 2008*, 2008, pp. 215–222.

[22] B. Mirkin, "Mathematical classification and clustering: From how to what and why," in *Classification, Data Analysis, and Data Highways*. Springer, 1998, pp. 172–181.

[23] I. S. Dhillon, "Co-clustering Documents and Words Using Bipartite Spectral Graph Partitioning," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2001, pp. 269–274.

[24] M. Rege, M. Dong, and F. Fotouhi, "Co-clustering Documents and Words Using Bipartite Isoperimetric Graph Partitioning," in *Sixth International Conference on Data Mining*. IEEE, Dec. 2006, pp. 532–541.

[25] S. Barkow, S. Bleuler, A. Prelić, P. Zimmermann, and E. Zitzler, "BicAT: a biclustering analysis toolbox," *Bioinformatics*, vol. 22, no. 10, pp. 1282–1283, May 2006.

[26] D. Filippova, A. Gadani, and C. Kingsford, "Coral: an integrated suite of visualizations for comparing clusterings," *BMC Bioinformatics*, vol. 13, no. 1, p. 276, Oct. 2012.

[27] M. Sun, C. North, and N. Ramakrishnan, "A Five-Level Design Framework for Bicluster Visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1713–1722, Dec. 2014.

[28] R. Santamaría, R. Therón, and L. Quintales, "BicOverlapper: A tool for bicluster visualization," *Bioinformatics*, vol. 24, no. 9, pp. 1212–1213, May 2008.

[29] P. Fiaux, M. Sun, L. Bradel, C. North, N. Ramakrishnan, and A. Endert, "Bixplorer: Visual Analytics with Biclusters," *Computer*, vol. 46, no. 8, pp. 90–94, Aug. 2013.

[30] M. Streit, S. Gratzl, M. Gillhofer, A. Mayr, A. Mitterecker, and S. Hochreiter, "Furby: fuzzy force-directed bicluster visualization," *BMC Bioinformatics*, vol. 15, no. 6, p. S4, May 2014.

[31] P. Xu, N. Cao, H. Qu, and J. Stasko, "Interactive visual co-cluster analysis of bipartite graphs," in *2016 IEEE Pacific Visualization Symposium*, Apr. 2016, pp. 32–39.

[32] Y. Onoue, N. Kukimoto, N. Sakamoto, and K. Koyamada, "Minimizing the Number of Edges via Edge Concentration in Dense Layered Graphs," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 6, pp. 1652–1661, Jun. 2016.

[33] M. Sun, J. Zhao, H. Wu, K. Luther, C. North, and N. Ramakrishnan, "The effect of edge bundling and seriation on sensemaking of biclusters in bipartite graphs," *IEEE Transactions on Visualization and Computer Graphics*, [to appear] 2018.

[34] J. Zhao, M. Sun, F. Chen, and P. Chiu, "BiDots: Visual Exploration of Weighted Biclusters," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 195–204, Jan. 2018.

[35] G. Y.-Y. Chan, P. Xu, Z. Dai, and L. Ren, "Vibr: Visualizing bipartite relations at scale with the minimum description length principle," *IEEE Transactions on Visualization and Computer Graphics*, [to appear] 2018.

[36] N. Pezzotti, J.-D. Fekete, T. Höllt, B. Lelieveldt, E. Eisemann, and A. Vilanova, "Multiscale visualization and exploration of large bipartite graphs," in *Computer Graphics Forum*, vol. 37, no. 3, 2018, pp. 549–560.

[37] A. Tanay, R. Sharan, and R. Shamir, "Discovering statistically significant biclusters in gene expression data," *Bioinformatics*, vol. 18, no. 1, pp. 136–144, Jul. 2002.

[38] M. Ailem, F. Role, and M. Nadif, "Co-clustering Document-term Matrices by Direct Maximization of Graph Modularity," in *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*, 2015, pp. 1807–1810.

[39] P. Riehmann, M. Hanfler, and B. Froehlich, "Interactive Sankey diagrams," in *IEEE Symposium on Information Visualization, 2005*, Oct. 2005, pp. 233–240.

[40] F. Bendix, R. Kosara, and H. Hauser, "Parallel Sets: Visual Analysis of Categorical Data," in *IEEE Symposium on Information Visualization*. IEEE, 2005, pp. 133–140.

[41] M. Bostock, V. Ogievetsky, and J. Heer, "D^3 - Data-Driven Documents," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011.

[42] NPashaP, "Viz - biPartite - default," http://bl.ocks.org/npashap/12cd547b1a3270603a139186b05415ff, 2018, [Online; accessed Sep-2018].

[43] Square, "Crossfilter," https://square.github.io/crossfilter, 2017, [Online; accessed Oct-2017].

[44] D. Steinböck, E. Gröller, and M. Waldner, "BiCFlows," https://users.cg.tuwien.ac.at/~waldner/bicflows/, 2018, [Online; accessed June-2018].

[45] A. Rind, D. Pfahler, C. Niederer, and W. Aigner, "Exploring media transparency with multiple views," in *Proceedings of the 9th Forum Media Technology 2016*. CEUR-WS, 11 2016, pp. 65–73.

[46] P. Salhofer, "MEHR! Medientransparenz," https://www.medien-transparenz.at, 2017, [Online; accessed Oct-2017].

[47] University of Trier, "dblp computer science bibliography," https://dblp.uni-trier.de/, 2018, [Online; accessed June-2018].

[48] C. North, "Toward measuring visualization insight," *IEEE Computer Graphics and Applications*, vol. 26, no. 3, pp. 6–9, May 2006.

[49] P. Saraiya, C. North, and K. Duca, "An insight-based methodology for evaluating bioinformatics visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 4, pp. 443–456, Jul. 2005.

[50] J. Brooke, "SUS - A quick and dirty usability scale," *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996.