Overlapping Clustered Graphs: Co-authorship Networks Visualization

Rodrigo Santamaría and Roberto Therón

University of Salamanca

Abstract. The analysis of scientific articles produced by different groups of authors helps to identify and characterize research groups and collaborations among them. Although this is a quite studied area, some issues, such as quick understanding of groups and visualization of large social networks still pose some interesting challenges. In order to contribute to this study, we present a solution based in Overlapper, a tool for the visualization of overlapping groups that makes use of an enhanced variation of force-directed graphs. For a real case study, the tool has been applied to articles in the DBLP database.

1 Introduction

Social network analysis has been a growing area of study in social sciences, mainly due to the amount of social information that can be recovered from internet social activities (blogs, chats, mail contacts, etc.) and from different public databases (movie and music databases, scientific article databases, etc.).

Information visualization has been extensively used by social scientists to aid in understanding these relationships. Most of the uses of information visualization in social networks are devoted to path-finding tasks, neighbor detection and most connected nodes (hubs) detection [9]. To perform these tasks, the usual visualization techniques are Node Links (NL) diagrams. NL diagrams represent entities as nodes (usually, points or small figures) and relationships as links (lines) that join related nodes. These diagrams are good for finding common neighbors and other characteristics, such as articulation points (nodes where two large subgroups join), but become cluttered and unreadable when the size of the graph is large [13]. Filtering and navigation through the graph must be implemented to dodge this problem. The primary alternate visualization technique to NLs are matrix graph representations, which perform well in finding most connected nodes and large complete subgraphs. Unfortunately matrix representations are not as good as NLs at conveying paths, and also have problems with large networks because of the space needed to represent the matrices (they are symmetric matrices, thus duplicating information, with lots of empty cells). The merging of both techniques is leading to promising results [12] although are still unable to deal with the large networks problem.

Some social data provides group information in addition to plain, individual relationships. This group information can help to simplify individual-level visualizations by taking them to group-level visualization, and it is key to understanding group relationships. In fact, research to find the best layout for NLs usually involves artificial classification of data by means of clustering or similar techniques, based on geometrical characteristics of the graph (usually path length between nodes). Most of these classification algorithms generate non-overlapping groups, which are useful for graph drawing but are not as good for group analysis, since real social groups are usually more complex, involving different degrees of overlap among groups.

There are techniques to find overlapping groups in data, such as fuzzy clustering [1] or biclustering [15], but there are also known overlapping groups in social data. Furthermore, some of the largest public databases contain information on social groups, such as IMDb (for movies) or DBLP (for scientific articles). In this paper, we present the application of a graph drawing method that speeds up the comprehension of these groups and exploits its use to simplify graph visualization. Section 2 presents related work in the area of social networks and clustered graph drawing. Section 3 explains the method to build overlapping group graphs, while Section 4 details its application to a real case study. Finally, Section 5 has our conclusions and summarizes some lines of future work that we are exploring.

Supplementary information, including more snapshots and a demonstration video with Smart Graphics' DBLP entries is available at http://vis.usal.es/artoverlapper.

2 Related Work

In this section, we will briefly survey the main social network tools and clustered graph drawing techniques in the present days.

2.1 Social network tools

There are a number of tools available to draw graphs, and the number of researchers that use these tools to analyze their data or as a starting point for their own graph implementations increase everyday. Some of these tools are mainly based in force-directed graphs, such as GraphViz [6] or Prefuse [11]. Force-directed graphs [8] make use of concepts such as gravitational or spring forces to layout nodes in a NL diagram.

These tools are usually more focused on aesthetics and usability, and they are used by a broad range of users, both scientific and non-scientific. For example, Prefuse, which is written in Java and it is accompanied by a comprehensive documentation and a large set of examples.

Other tools, such as Pajek [3] or JUNG (http://jung.sf.net) are focused on statistical analysis and drawing methods. Contrary to GraphViz or Prefuse, Pajek and JUNG are used mainly by specialists in social sciences and graph drawing.

2.2 Clustered Graph drawing

Clustered graphs (CGs) are NL representations where groups or zones of related nodes are highlighted (specially colored, for example). These representations use non-overlapping clusters present in the data or obtained by clustering techniques (usually hierarchical clustering).

Three main types of CG drawings have been identified [2]: hierarchical clustered graphs, compound graphs and force-directed clustered graphs.

Hierarchical clustered graphs, introduced by Eades and Feng [5], start by drawing the highest level of a hierarchical clustering (only one cluster for all the nodes), and then go on drawing in decreasing z coordinates additional graphs with lower levels of clustering, where nodes are clusters and edges join clusters that were together in the upper clustering (see Fig 1a).

Compound graphs [19] are Hierarchical Clustered Graphs in which the inclusion relationship is taken into account to draw the hierarchical clustering in a single graph representation. The final visualization is very similar to a Treemap [18] (see Fig. 1b).

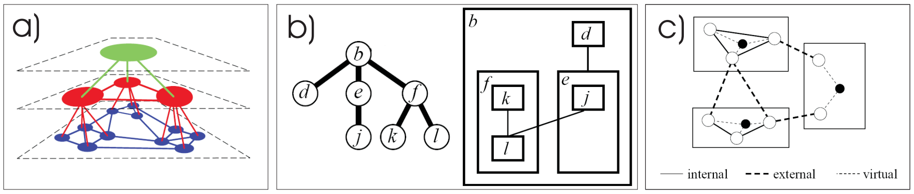

Fig. 1. a) Hierarchical clustered graph. A 3D visualization with different levels of clustering. Edges relate clusters together in the upper level. b) Compound graph, the cluster hierarchy at the left is translated to a graph layout of inclusion clusters. c) Force-directed clustered graph. Edges internal and external to clusters act as spring forces, with additional help of virtual edges by using virtual nodes in each cluster (all these figures are taken from [2]).

Finally, force-directed clustered graphs (FDCGs) are the most widespread Clustered Graphs. A combination of spring forces for a single clustering is used: inter-cluster, intra-cluster and (sometimes) ancillary forces by using virtual nodes in each cluster. In addition, a gravitational repulsion between each pair of nodes is applied (see Fig.1c).

Most of the social network tools discussed above have been used to implement FDCGs. For example, SocialAction [16] uses the Prefuse visualization kit and *betweeness* centrality (a measure of the relative importance of a node within a graph) to determine and draw clusters, simplifying the visualization of graph drawings. Vizster [10] is also based on Prefuse and group zones by clustering, allowing the user to define its granularity. Frishman and Tal [7] take GraphViz as a starting point for a dynamic drawing of clustered graphs. It is common in this implementations that the clustering displayed in the graph is a level of a hierarchical clustering, that can be changed at user's demand.

Besides Compound Graphs, where intersection between groups is reduced to inclusion, none of these graph drawing methods deal with overlapping groups, that are usually present in real data, or which can be obtained by newer classification techniques such as biclustering. Overlapping groups are usually a better way to display connections between entities, avoiding the threshold cut in hierarchical clustering that can assign doubtful nodes to a group.

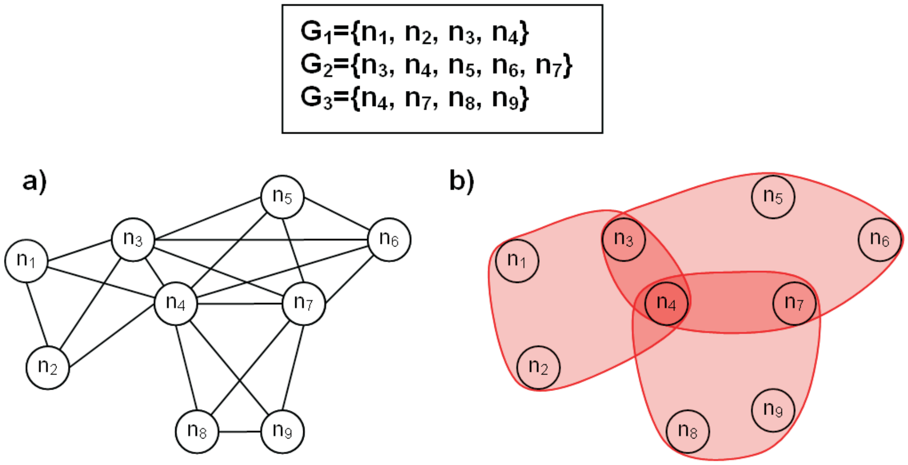

Fig. 2. a) Three groups are represented as complete subgraphs, with edges between all their members. b) Edges are hidden and replaced by transparent hulls wrapping the elements in each group. The relationships between groups arise quickly and elements like n_4, present in the three groups, are highlighted by hull overlapping.

3 Group drawing with Overlapper

In this section we describe the drawing methods used by the presented visualization technique, focusing on graph building, layout and interaction.

3.1 Overlapper

Overlapper is a tool designed for visual analysis of data from different fields, such as social groups or biclustering results. It integrates and links different ancillary visualization techniques, such as parallel coordinates, scatter plots, tree maps and node-link diagrams to gain insight into the field of study. The main visualization in Overlapper is based in the graph that is described below.

The overall structure of the tool is redesigned for each field of study to fit with the specific characteristics of the data (e. g. node types, group types, relevant ancillary visualization techniques). Two versions have been developed, one for movie world analysis [20], awarded at the 14th Graph Drawing Contest [4], and another one for biclustering results analysis [17].

3.2 Group Building

Graph drawing with Overlapper centers on groups and their representation. To achieve this, the data to be represented should contain information on groups, either from previous information (as could be the case of databases like IMDb or DBLP) or from classification techniques (producing either overlapping or non-overlapping groups). No matter what the source of data, we produce a list of groups $G_1,...,G_k$ each one containing elements, that will be treated as nodes in a graph: $G_i = \{n_{i1},...,n_{ik}\}$. Note that, for some groups, it is possible that $G_i \bigcap G_j \neq \oslash$.

From this list of groups, each G_i is represented as a complete subgraph, with the resulting graph being the union of all these subgraphs (see Fig. 2). Nodes and edges in this graph $G = (N,E)$ correspond to:

$$N = \{n_i | \; \exists \; G_k \; with \; n_i \in G_k\} \tag{1}$$

$$E = \{(n_i,n_j) | \; \exists \; G_k \; with \; n_i,n_j \in G_k\} \tag{2}$$

3.3 Graph layout and drawing

The graph is displayed using a force-directed layout, with two kinds of forces, in a way similar to other force-directed graphs, such as the ones implemented in the social network tools discussed in Section 2. A spring force S attracts nodes joined by an edge, while a gravitational force X repulses each node from every other node (see eq. 3). Both forces depends on the distances among nodes. The S force is kept stronger than X to avoid dispersion of groups. The overall result is that nodes in the same groups tend to be closer and are separated from nodes in different groups.

$$F_i = \sum_{(n_i,n_j)\in E} S_{i,j} + \sum_{n_j \in N} X_{i,j} \tag{3}$$

The layout is computed iteratively, so after each cycle, nodes are relocated depending on the applied forces, and forces are again recomputed for the new locations of nodes.

For each layout cycle, nodes are drawn as circles at their recomputed positions. In addition, for each group a rounded transparent shape (hull) is drawn, instead of drawing their edges. The contour of the hull is determined by the positions of the outermost nodes in each subgraph, that are taken as anchor points for a closed spline. The outermost nodes are computed on-the-run by checking the positions of the nodes in each group, and determining which are the ones in the periphery at each moment (those with the minimum or maximum x,y coordinates).

The transparency level of hulls is determined by the maximum number of overlapping groups in a determinate set of groups, n_max. If 0 is the transparency level of transparent colors and k is the transparency level of solid colors, the transparency of each hull is $(k - k_0)/n_max$. k_0 is a low value that keeps the maximum overlap from being fully solid.

Hull drawing, although based on edges, does not clutter the visualization. The transparency of hulls makes intersecting zones among groups more opaque, thus highlighting the highly connected, hub-like zones.

Finally, to boost comprehension of the relationships among groups, intersecting nodes are drawn as pie charts, with as many sectors as groups the node belongs to. This way, after getting used to our method, the analysis of group interactions becomes

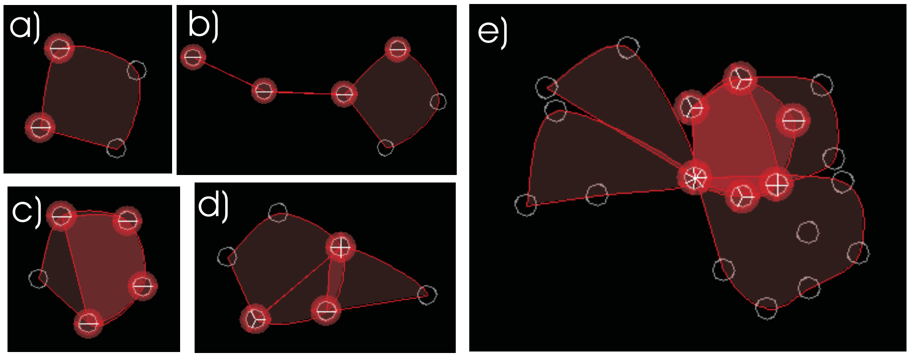

Fig. 3. Some examples of real group relationships from DBLP. Each hull is a group (an article) and each node an author. a) Three articles, in the first one four authors collaborated, the other two were written by single authors. b) A chain of scientific collaborations. Most left author has written two articles, one alone and another one with the chain's following author that, in turn, has written another article with the following author in the chain. Finally, this author also collaborated in a paper with three other colleagues, one of which has written an article alone. c) Two articles written by almost the same people except one person. d) Four articles. The most prolific author worked alone once, another time with just a colleague and two more times with groups of two and four other people. e) A more complex interrelationship of authors. The most relevant author, present in all (seven) publications is quickly identified. Also, the most prolific authors worked together on a couple of papers.

easy and unambiguous, and identifying the most connected nodes, thanks to transparent hulls and pie charts, is quicker (see Fig. 3). Note that for groups of two elements, hulls are drawn as lines, and for groups of one element, no hull is drawn at all, so in these cases piecharts are used to distinguish, for example, a member of a group that has worked in other projects, from a member that just worked in this group. These very small groups can occur in research paper datasets (some papers are authored by a small number of researchers).

3.4 Graph interaction

Once the graph is built and displayed, it can be manipulated by the user in a number of ways. Regarding to the layout, the user can change the parameters X and S and modify the representation by dragging and fixing node positions. Regarding the graph drawing, the user can visualize or hide nodes, edges, hulls and pie charts; draw labels of node and group names; and highlight the nodes connected to a particular node. In order to facilitate the navigation through the graph, the user can search for author names, filter low related groups, and get an overview of the complete graph. Finally, the user can export the graph visualization to different image formats.

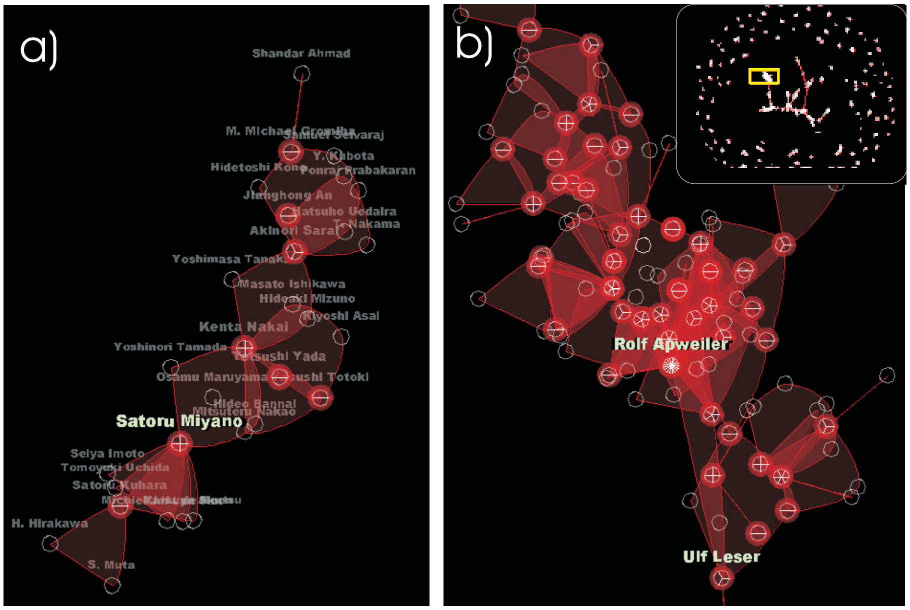

Fig. 4. a) Japanese researchers in a peripheral group of authors that have published in Bioinformatics. Although it is a more complex group than those presented in fig. 3, interactions between authors are clear. b) Top-left part of the central subgroup, where the most influent Bioinformatics' contributors are present. This zone is mainly populated by German authors. Top-right square shows the overall view of the complete graph, with disconnected, peripheral groups surrounding the central group, with clear branches.

4 Case Study

In order to demonstrate the capabilities of our representation, we have used subsets of the DBLP article database [14]. Specifically, we have focused on the articles from the journal *Bioinformatics*, with 10 years of publications (since 1998) and over 3500 articles and 25000 authors.

The first five years of *Bioinformatics*, with 932 articles and 2197 authors, represents a good challenge for our visualization tool. Filtering all the articles not related to any other, we reduced the number of articles and authors to 615 and 1212, respectively. The graph layout, directed by forces, disperses unrelated articles around the visualization space, leaving the highest related subgroup in the center of the visualization (see Fig. 4b, top-right square).

It is remarkable that the nationality of the authors is reflected in the way research groups are formed and publish. For example, in Fig. 4a we observe that a large peripheral group is formed almost exclusively by Japanese researchers.

The central group, with the most influential authors in *Bioinformatics* in its first five years, also include nationality groupings (see Fig. 5a for Russian researches, interconnecting with German colleagues of Fig. 4b).

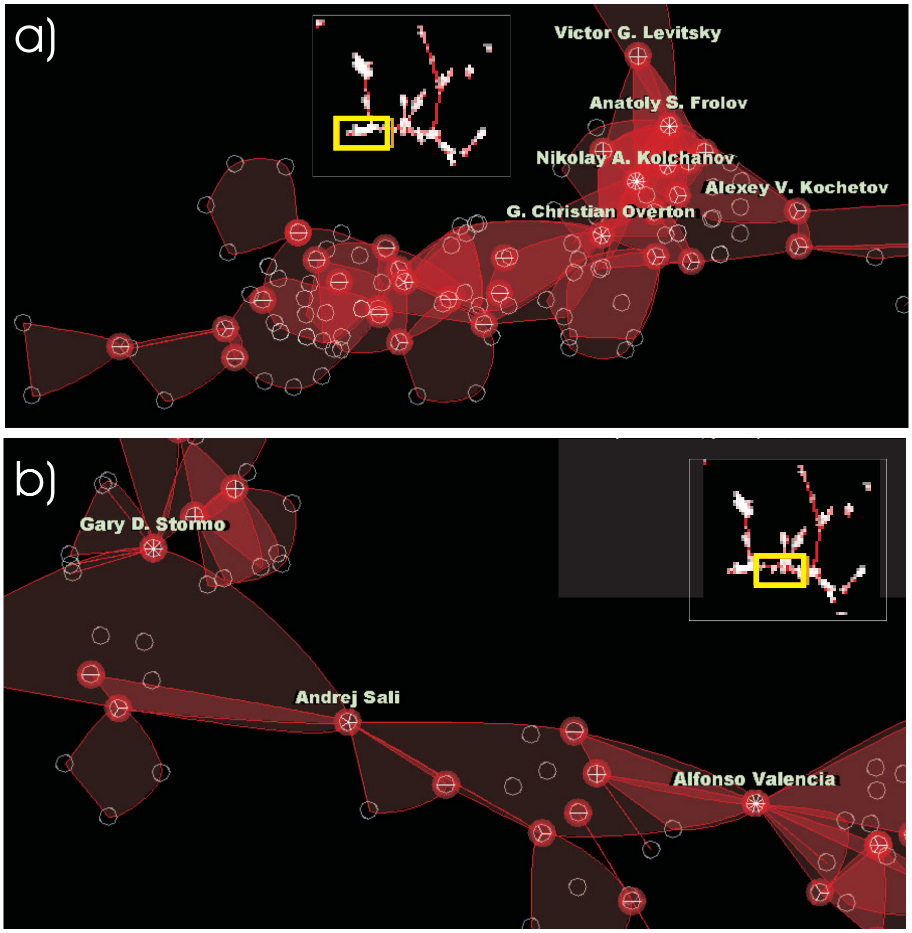

Fig. 5. a) Bottom left branch of the central main group. Most of the authors are Russian, with hub figures as Nikolay A. Kolchakov and articulation points as Victor G. Levitsky and Alexey V. Kochetov. b) Central part of the main group. Here, Alfonso Valencia reveals as one of the most connected nodes and also as an articulation point. Other articulated and connected authors are Gary D. Storno and Andrej Sali.

Although complexity in group interactions arises in the groups of Fig. 4 and Fig. 5 with respect to the ones in Fig. 3, relationships are still clear. Due to forces equilibrium, on few occasions one node can be placed too close to a group that contains nodes related to it, but the node itself does not pertain to the group. In these cases, pie charts disentangle possible ambiguities. Hub nodes and articulation points are identified quickly, as can be seen in Fig. 5b. The identification of hub nodes is a difficult task in NL diagrams [12]; and it is mainly solved by this visualization.

Finally, we must consider that article graphs are usually sparse, and therefore cluttering is less frequent than in other denser graphs. However, the use of hulls instead of edges significantly simplifies the comprehensibility of the visualization whereas the traditional graph drawings of nodes and edges are unreadable even for these relatively simple examples (see Fig. 6).

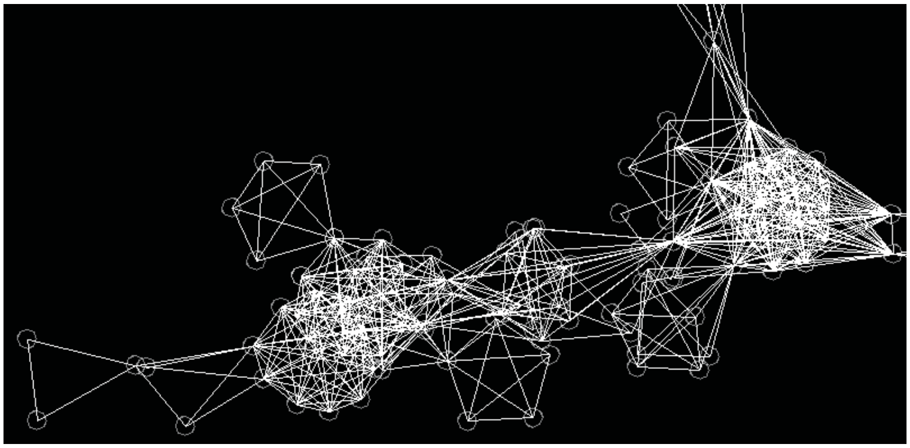

Fig. 6. Traditional NL representation of fig. 5a. The visualization becomes cluttered and details about groups are unreadable.

5 Conclusions and future work

We have developed a new graph drawing method based on a different way of representing relationships among nodes. Individual relationships are taken as the basic infrastructure for the graph layout, but are hidden to the benefit of group relationships. The resulting graph drawing method is only applicable if group information is present or can be inferred with some classification technique. In these cases, which are common in social networks, the overlapping clustered graph drawing has advantages over standard NL diagrams. The edge cluttering is avoided and is substituted by an unoffensive hull overlapping, that is exploited to highlight intersecting groups.

The presented case study demonstrates that our method can successfully deal with large sparse graphs without losing readability and provides quick insight into group relationships. Also, thanks to the pie charts and the force-directed graph layout, the identification of hub nodes and articulation points is enhanced.

The use of this technique in denser graphs is under research. The overlapping display method will need modifications in the layout algorithm to deal with possible misplacing of nodes inside group hulls in which they are not included. This issue is solved by the force-directed layout for sparse graphs, but becomes a problem in denser ones.

6 Acknowledgments

This work was supported by the Ministerio de Educación y Ciencia of Spain under project TIN2006-06313 and by a grant from the Junta de Castilla y León autonomous government.

References

1. A. Baraldi and P. Blonda. A survey of fuzzy clustering algorithms for pattern recognition. *IEEE Trans. on Systems, Man, and Cybernetics*, 29(6):786–801, 1999.
2. R. Brockenauer and S. Cornelsen. Drawing clusters and hierarchies. In *Drawing Graphs, LNCS*, volume 2025/2001, pages 193–227, 2001.
3. W. de Nooy, A. Mrvar, and V. Batagelj. *Exploratory Social Network Analysis with Pajek*. Cambridge University Press, New York, 2005.
4. C. A. Duncan, S. G. Kobourov, and G. Sander. Graph drawing contest report. Technical report, 2007.
5. P. Eades and Q.-W. Feng. Multilevel visualization of clustered graphs. In *Graph Drawing, LNCS*, volume 1190/1997, pages 101–112, 1996.
6. J. Ellson, E. Gansner, L. Koustsofios, N. Stephen, and G. Woodhull. Graphviz – open source graph drawing tools. In *Graph Drawing, LNCS*, volume 2265/2002, pages 483–485, 2002.
7. Y. Frishman and A. Tal. Dynamic drawing of clustered graphs. In *Infovis*, pages 191–198, 2004.
8. T. M. J. Fruchterman and E. M. Reinhold. Graph drawing by force-directed placement. *Software – Practice and Experience*, 21:1129–1164, 1991.
9. M. Ghoniem, J.-D. Fekete, and P. Castagliola. On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. *Information Visualization*, 4:114–135, 2005.
10. J. Heer and D. Boyd. Vizster: visualizing online social networks. In *IEEE Symp. on Information Visualization*, page 5, 2005.
11. J. Heer, S. K. Card, and J. A. Landay. Prefuse: a toolkit for interactive information visualization. In *SIGCHI Human Factors in Computing Systems*, pages 421–430, New York, NY, USA, 2005. ACM Press.
12. N. Henry and J.-D. Fekete. Matlink: Enhanced matrix visualization for analyzing social networks. In *Human-Computer Interaction*, pages 288–302, 2007.
13. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Trans. on Vis. and Comp. Graph.*, 6(1):24–43, 2000.
14. M. Ley. The dblp computer science bibliography: Evolution, research issues, perspectives. In *9th Intl. Symp. on String Processing and Information Retrieval*, pages 1–10, 2002.
15. S. Madeira and A. Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Trans. of Computational Biology and Bioinformatics*, 1(1):24–45, 2004.
16. A. Perer and B. Shneiderman. Balancing systematic and flexible exploration of social networks. *IEEE Trans. on Vis. and Comp. Graph.*, 12(5):693–700, 2006.
17. R. Santamaría, R. Therón, and L. Quintales. Bicoverlapper: A tool for bicluster visualization. *Bioinformatics*, 24(9):1212–1213, May 2008.
18. B. Shneiderman and M. Wattenberg. Ordered treemap layouts. In *Infovis*, pages 73–78, 2001.
19. K. Sugiyama, K.; Misue. Visualization of structural information: automatic drawing of compound digraphs. *IEEE Trans. on Systems, Man and Cybernetics*, 21(4):876–892, 1991.
20. R. Therón, R. Santamaría, J. García, D. Gómez, and V. Paz-Madrid. Overlapper: movie analyzer. In *Infovis Confererence Compendium*, pages 140–141, 2007.