

Towards Unambiguous Edge Bundling: Investigating Confluent Drawings for Network Visualization

Benjamin Bach, Nathalie Henry Riche, Christophe Hurter, Kim Marriott, Tim Dwyer

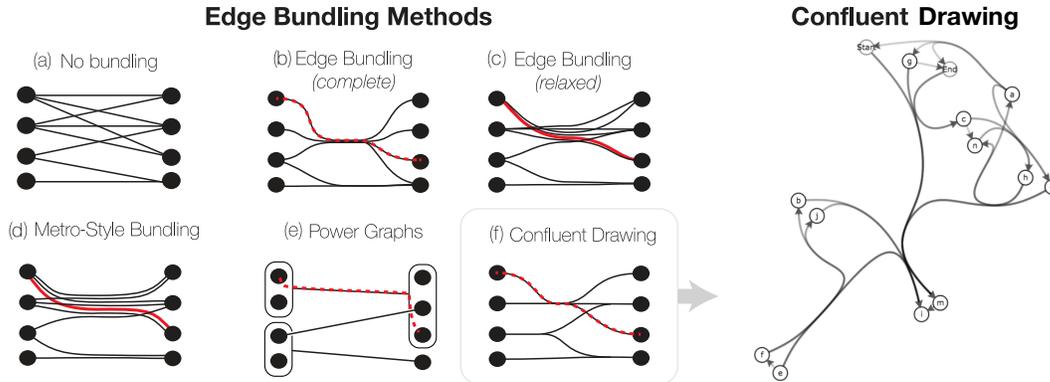


Fig. 1. Techniques designed to reduce edge clutter (left) and example of confluent drawing for a directed network (right).

Abstract— In this paper, we investigate Confluent Drawings (CD), a technique for bundling edges in node-link diagrams based on network connectivity. Edge-bundling techniques are designed to reduce edge clutter in node-link diagrams by coalescing lines into common paths or bundles. Unfortunately, traditional bundling techniques introduce ambiguity since edges are only bundled by spatial *proximity*, rather than network *connectivity*; following an edge from its source to its target can lead to the perception of incorrect connectivity if edges are not clearly separated within the bundles. Contrary, CDs bundle edges based on common sources or targets. Thus, a smooth path along a confluent bundle indicates precise connectivity. While CDs have been described in theory, practical investigation and application to real-world networks (i.e., networks beyond those with certain planarity restrictions) is currently lacking. Here, we provide the first algorithm for constructing CDs from arbitrary directed and undirected networks and present a simple layout method, embedded in a sand box environment providing techniques for interactive exploration. We then investigate patterns and artifacts in CDs, which we compare to other common edge-bundling techniques. Finally, we present the first user study that compares edge-compression techniques, including CD, power graphs, metro-style, and common edge bundling. We found that users without particular expertise in visualization or network analysis are able to read small CDs without difficulty. Compared to existing bundling techniques, CDs are more likely to allow people to correctly perceive connectivity.

Index Terms—Network visualization, edge compression, confluent, power graph, bundling

1 INTRODUCTION

Edge-bundling techniques are designed to reduce clutter in “node-link” diagrams (Fig. 1(a)). Currently bundling involves identifying subsets of edges following similar trajectories and grouping them together into bundles (spatial bundling). On one hand, researchers argue that bundles provide a good overview of the connectivity in large or dense graphs, as the clutter caused by edge crossings is greatly reduced; and the thickness and shape of these edge bundles provide visual clues for high-level connectivity.

On the other hand, bundling techniques may be misleading. Each bundle involves a subgraph induced from a set of edges that are spatially nearby, in other words, the particular edge-set associated with each bundle can be rather arbitrary and related to layout rather than graph topology. This arbitrariness can lead to ambiguities in precisely perceiving connectivity information encoded in the bundle. For a given edge that joins a bundle, it can be impossible to see precisely where that same edge leaves the bundle, and therefore to which node it connects (Fig. 1(b)). Thus, with spatial edge-bundling techniques, precise connectivity is lost when bundles are created.

In this paper, we investigate *Confluent (Edge) Drawings* (CD), a technique for bundling edges based on network connectivity, and without any loss of information [9]. In a CD, two edges are bundled only if both source nodes are connected to both target nodes (Fig. 1(f)), i.e. the reader can follow edges similar to following railway tracks—singly, or combined into bundles—from source to target as long as the path does not require a sharp turn. In contrast to previous bundling techniques, CD may require some training to confidently follow edge paths through bundles, but connection ambiguities are eliminated. Previous research into CD has investigated theoretical aspects of CDs with a particular focus on identifying classes of networks for which there is a planar CD representation, i.e. one with no crossings. Currently there is no practical solution available that creates a (possibly non-planar) CD representation for arbitrary networks and there has been no study of their applicability to real-world networks, or their readability by people.

- Benjamin Bach is with Microsoft Research-Inria Joint Centre, Saclay, France (benj.bach@gmail.com)
- Nathalie Henry Riche is with Microsoft Research, WA, USA (nath@microsoft.com)
- Christophe Hurter is with ENAC, Toulouse, France (christophe.hurter@enac.fr)
- Kim Marriott is with Monash University, Melbourne, Australia, (kim.marriott@monash.edu)
- Tim Dwyer is with Monash University, Melbourne, Australia (tim.dwyer@monash.edu)

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxx/TVCG.201x.xxxxxx

Our research aims at expanding this knowledge for practical use of CD in network visualization. Specifically, this article provides the following contributions:

- an algorithm to compute directed/undirected confluent bundles and to lay out nodes and edge bundles for any network (beyond confluent-planar or other restrictions);
- an approach to evaluate CD and edge-bundling techniques based on the systematic analysis of visual patterns produced for a corpus of network motifs and the discussion of visual artifacts;
- a controlled user study assessing the readability of CD compared to existing edge-bundling techniques (with [37] and without ordered path separation [29]), and the state-of-the-art “edge compression” technique [13] replacing edges by node groupings without loss of connectivity information.

The full study material and illustrative examples of networks visualized with CD, as well as an interactive Confluent Design Sand Box can be found online: <http://confluentgraphs.benjbach.me>.

2 BACKGROUND AND RELATED WORK

2.1 Edge Congestion and Visual Clutter

Densely connected networks have many more edges than nodes. Node-link representations of such networks that use straight-lines to represent the edges quickly become cluttered with masses of lines crossing each other and also crossing node glyphs (Fig. 1(a)). This problem is referred to as *edge congestion* [8] and it seriously limits the scalability of node-link representation to complex real-world networks. However, node-link representations are still the most common representation for networks and are easily understood by people, compared to, for example adjacency matrices.

One way to reduce edge-congestion is to replace groups of highly connected nodes with a single *meta-node* in the node-link diagram. For example, Dunne and Shneiderman [11] greedily identify cliques and other motif structures to be collapsed into meta-nodes drawn with distinctive glyphs. Doing so may ellide many edges and therefore edge-congestion from the drawing and the result may still convey high-level structure. However, individual nodes are hidden in the process and interaction is required in order to allow the user to recover this information. Other methods explore interactively navigable hierarchical clusterings to provide multiple levels of overview and detail [2, 4, 5].

While there are many types of interaction that can ultimately assist the user in understanding precise connectivity of a complex network on demand, any particular visualization of a set of nodes should strive to convey connectivity between those nodes as faithfully as possible [34]. We therefore focus on techniques that try to provide a clutter-free view of the connectivity, regardless of the possibility of interaction.

2.2 Edge Bundling

One of the most employed non-interactive method to reduce edge clutter is edge bundling. Edge bundling works by grouping edges by spatial *vicinity* (we call these techniques *spatial edge bundling*) and create curves that are easy to follow with the eye, mimicking bundles of wires (Fig. 1(b-d)). Edge bundling was first introduced by Holten for hierarchical graphs, to bundle edges between different subgraphs [24]. Since then, a plethora of applications and optimizations has been investigated, well summarized by Zhou et al. [45]. Common extensions and related techniques include, bundling undirected [26] graphs in force-directed layouts, directed graphs [40], directed and multidimensional [35], bundling on geographic maps [36], as well as dynamic networks [28], to name just a few.

The problem with *spatial bundling* techniques is that bundles can lead to the perception of connections between unconnected nodes since individual connections many visually be hard to distinguish if bundled. This problem has previously be named *edge ambiguity* [32] and several solutions have been proposed: relaxing the bundling by linear interpolation or interactive lenses [44]. Pupyrev *et al.* [37] introduced a “metro-style” bundling technique, routing edges carefully around nodes within bundles as parallel segments such that all individual edges remain visible (Fig. 1(d)). On very careful inspection each edge path can

still be followed from source to destination, however complex networks can be appear cluttered.

Other techniques are very restrictive about the amount of bundling that they allow in order to limit ambiguity. For example, FlowMaps [36] bundle edges of a single node only, resulting in a spanning-tree starting at the node of interest. Luo et al. [32] developed a more aggressive greedy heuristic for bundling that avoids ambiguity by selectively bundling edges. However, this ambiguity-free edge bundling is still based on visual metrics, such as the closeness of two edges and the degree of bundling may be very limited in dense graphs. Confluent drawings, described long before, bundle *all* edges between nodes that are non-ambiguous. In other words, a confluent graph drawing finds the maximal set of non-ambiguous bundles in a given network.

Despite the growing corpus of research, few insights have been reported on the readability on edge-bundling techniques based on actual user testing. While techniques have been presented with selected images, objective reports on quantitative quality measures and use cases that highlight how a specific technique helps to extract relevant information from a network are lacking. To the best of our knowledge the work presented here is the first that explicitly involves users into an evaluation. Our study, involving 15 participants without prior knowledge of visualization, compares common edge-bundling techniques for path-following tasks and sheds some light on the individual strengths and weaknesses of each approach.

2.3 Power-Graph Edge Compression

Power-graph edge compression is an alternative to spatial edge bundling that creates bundles based on connectivity of nodes. *Power Graphs* use a hierarchical aggregation (or decomposition) of the nodes to reduce edge congestion while still displaying the full set of nodes and conveying precise connectivity between them. This is done by surrounding groups of nodes with closed curves as in Figure 1(e). If the nodes inside the group all have a common neighbor outside the group, the set of edges from those constituent nodes to the common neighbor can be unambiguously replaced with a single (power) edge connected to the group’s enclosing curve. A power edge connecting a group with m nodes to another group with n nodes replaces $m \times n$ edges in the original graph. For example, Fig. 2(b) shows a power graph where power edges replace the original edges from the uncompressed graph in Fig. 2(a).

Note, that the goal of power-graph decomposition is quite different to the kind of decompositions used to create groups around highly connected parts of the graph, i.e. *communities* [33]. Such community-based clusters seek to maximise the density of edges within a group, rather than between, and so offer little precise information about connectivity that can be used in a lossless compression.

Power graphs have been previously used in biology [39]. Computing a power-graph decomposition that minimizes the total number of edges turns out to be a challenging optimization problem, however an effective heuristic is given by Dwyer et al. [12]. Dwyer et al. [13] further present results from a controlled user study showing that shortest-path task completion time with power graphs is significantly faster compared to regular node link diagrams.

2.4 Confluent Drawings

Similar to power graphs, *Confluent (edge) drawings* (CDs) avoid ambiguities in traditional edge bundling by taking graph topology—i.e. connectivity—into account when creating bundles. Two edges are bundled only if both source nodes are connected to both target nodes (Fig. 1(f)). The bundles are then drawn using smooth curves and acute angles between bundles, such that the reader can be sure that a smooth path connecting nodes indicates unambiguously that they are connected [42].

Introduced in the Graph Drawing community a little over a decade ago, confluent drawings were not originally intended to be a generic representation for all networks. Rather they were introduced as a method for increasing the class of undirected networks that could be drawn without edge crossings [9]. Thus, virtually all prior research has focussed on *planar* confluent drawings. In particular, there has

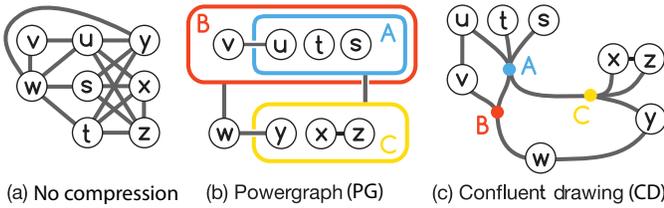


Fig. 2. Logical equivalence of power graph (b) and confluent edge drawing (c) for the same networks (a). Both compression techniques preserve all connectivity information.

been research identifying classes of networks that have or do not have a planar confluent drawing [9, 15]; investigating the theoretical complexity of determining if such a planar confluent drawing exists [17, 27]; giving heuristic algorithms to determine if such a drawing exists [9, 23]; giving algorithms to compute planar confluent drawings for restricted classes of networks or confluent drawings [23]; computing orthogonal routes for a (planar) confluent topology [38]; rotary-glyphs to draw cliques [9]; and extending the notion of confluence to other kinds of networks such as layered networks [16] and Hasse diagrams [18].

The question of how to extend these techniques and the associated theory to non-confluent-planar input graphs has, to our knowledge, so far been avoided. This limits their applicability to real-world graphs, where typically no such guarantee is available. It is not surprising then that other practical issues concerning the utility of CD for real-world network analysis remain unexplored. With our research involving a systematic comparison and a user study, we begin to close this gap by addressing the question of whether confluent drawings (extended to include non-planar confluent drawings) are a practical and useful technique for drawing arbitrary networks and what are the implications for design and directions for future research.

3 BUILDING AND RENDERING CONFLUENT EDGE DRAWINGS

We now describe how to create confluent edge drawings for any input graph using power-graph decomposition, with consideration of how to layout nodes and render bundles.

3.1 Leveraging Power-Graph Decomposition

We propose to build confluent drawings (CD) from power-graph (PG) decomposition, as used in power-graph edge compression (see Sec. 2.3). Fig. 2 illustrates how a CD can be derived straightforwardly from a PG. The CD in 2(b) is derived from original graph in the 2(a) using the `cola.js` [1] implementation of the greedy beam-search power graph decomposition method described in [12].

The same decomposition method is used in all examples in this paper and involves precisely four steps:

- 1) Each group in the PG corresponds to a bundle junction in the CD, as indicated by similarly labelled groups and junctions in the figure.
- 2) Leaf children of each PG group are connected in the CD to the junction corresponding to the group, e.g. nodes s, t and u are children of group A in the PG and hence in the CD s, t, u are connected by a smooth curve to junction A .
- 3) Junctions corresponding to groups that are contained within another group are connected to that parent group’s corresponding junction, e.g. group A is a child of group B in the PG, hence, in the CD, junctions A and B are connected.
- 4) Finally, power edges connected to groups correspond to additional connections to junctions, e.g. power edge $A-C$ becomes a connection between junctions A and C and the edge from node w to group B becomes a connection in the CD from node w to junction B . Edges between leaf nodes in the PG remain unchanged in the CD: $u-v$ and $x-z$.

Conversion of power graphs with directed edges works similarly. The only difference is that we create—as necessary—two junctions for each group, one for incoming and one for outgoing edges.

While confluent and power graphs are logically equivalent, their renderings are obviously not visually equivalent. On one hand, the

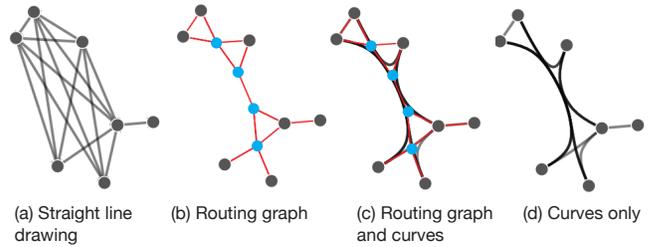


Fig. 3. Layout and curve drawing without node-splitting.

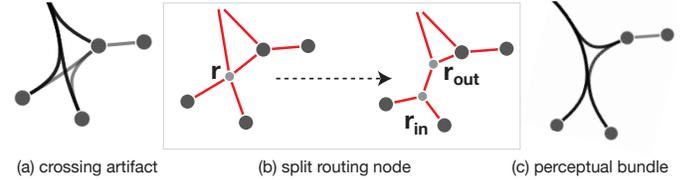


Fig. 4. Splitting nodes in the routing graph to avoid visual artifacts.

power-graph decomposition technique reduces edge crossings by introducing nested enclosing shapes to represent modules. However, as noted in the Power-Graph readability study by Dwyer et al. [13], the level of nesting is a source of error. Also, it seems that without training, some people assume the enclosed groups indicate densely connected clusters rather than sets of nodes with structural equivalence.

It may be that a Confluent Drawing of the PG structure offers a more intuitive representation of hierarchical connectivity than the usual PG rendering with enclosing regions for each group. For example, smooth CD edge bundles support the Gestalt principle of Continuity, considered important for network visualisation by Ware et al. [42]. However, replacing containment by bundled curves may introduce additional edge crossings and it is not clear how these will affect readability.

The following section describes the approach we use to create a CD layout and rendering from the set of junctions derived from the PG groups, as described above, in order to explore these considerations.

3.2 Rendering Confluent Edge Drawings

After having created the power graph for an arbitrary network, there are two main aspects for rendering confluent drawings: (i) finding a node layout that optimizes for curve readability, and (ii) drawing links as curves to bundle them. Both steps are detailed in the following and illustrated in Fig. 3.

Node Layout

To yield readable curves, we need to take curve positions into account when creating the graph layout. Therefore, we first create a *routing graph* containing the nodes of the initial graph plus a *routing node* \bullet for every junction of two or more bundles (Fig. 3(b)). Routing nodes are obtained from the modular decomposition in that every module m with nodes or sub-modules M_m becomes a joint node for the edges connecting the elements in M_m to their neighbors. For every $n \in M_m$, the routing graph contains a *routing edge* $e(n, m) \in RE$ (Fig. 3, \rightarrow).

We can now lay this graph out, for example, using a force-directed algorithm. Fig. 1 shows an example of a social network arranged with the `cola.js` [1] force-directed layout with additional “non-overlap” constraints to keep nodes and junctions well separated. Except where otherwise noted, this is the layout method used in all of our examples in our study.

Link Rendering

Curves are created for each edge, along the shortest paths in the routing graph starting from the edge’s source s to its target t . Routing points not being part of any shortest path, are removed subsequently. We decided to draw curves using common B-splines and to use the routing nodes

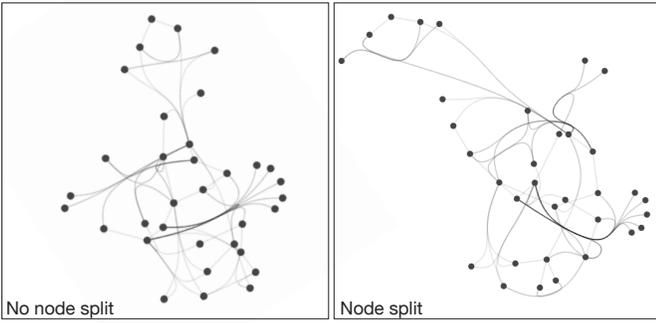


Fig. 5. Difference in confluence drawing with and without node-splitting. Node-split creates less tangled graphs.

along the path from s to t through the routing graph as control points for the splines. The resulting image is shown in Fig. 3(c).

However, this way of rendering confluent graphs, results in a specific artifact of crossing links where semantically there is a bundle (Fig. 4(a)). This happens when a routing point r has more than one incoming routing edge $|RE_{in}| > 1$ and more than one outgoing routing edge $|RE_{out}| > 1$. We solve this case by replacing each routing node r into two routing nodes r_{in} and r_{out} . r_{in} is attached to only the incoming edges RE_{in} and r_{out} is attached to only the outgoing edges RE_{out} . r_{in} and r_{out} are connected with a new routing edge (*node-splitting*, (Fig. 4(b)). By splitting certain routing nodes, we obtain bundles as shown in Fig. 5(b) instead of bundles with crossing artifacts as shown in Fig. 5(a).

3.3 Design Sandbox

Being able to render arbitrary non-planar networks as CDs, we now describe drawings generated by our algorithm for real-world networks. To explore several design alternatives, we implemented a *Confluent Design Sandbox* with the following features:

- control the *opacity and thickness of edges* to convey network density and the number of edges in bundles;
- control the *level of relaxation of bundles*: interpolating from straight edges to complete CD bundles;
- *mouse over nodes* to highlight their relations and neighbors;
- *drag and fix nodes* to manually refine the layout;
- encode directed links with *arrows or animated moving particles*.

Fig. 6 shows an authorship network with people connected to documents they have authored. Since CD is based on power-graph decomposition, it usually produces salient patterns for such bipartite structures. In the initial CD layout, some bundles appear to overlap or cross each other, inducing visual clutter. To detangle some of the bundles, we manually refined the layout by pinning the central node and dragging others further apart. Salient sub-structures became more visible, e.g. in Fig. 6 we see: (a) sets of authors who have only collaborated with the main author on a single publication; (b) the closest co-author identifiable by darker bundles (indicating high density bundles), similar to the main author; and (c) papers that are shared by three central authors which are identified by hovering over the nodes.

A second example, Fig. 7, is a directed communication network depicting retweets: a directed edge from person A to person B indicates that B shared information that was initially shared or created by A (B retweeted a tweet from A). Such directed networks are particularly interesting to represent with CD, compared to node-link diagrams, as bundles are split by their directionality. We found that directed CDs benefitted from using animated particles flowing along the trajectories of bundles and links to depict their direction (similar to [25]); especially if networks are dense and arrows are hard to perceive. The animation can be viewed on our website¹. Structures such as central nodes that many people retweet (sources) and nodes that aggregate information by retweeting many others (sinks) are also illustrated in Fig. 7. From our observations with CD we can conclude that using animated particles with some degree of relaxation makes it easier to identify these types of patterns, and may help novices to familiarize themselves with CDs.

¹http://aviz.fr/~bbach/confluentgraphs/#scenario_twitter

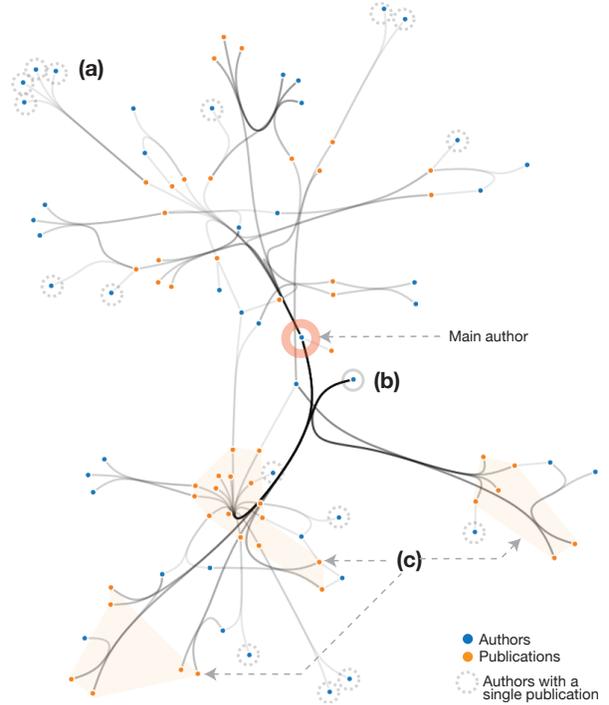


Fig. 6. Ego-centric authorship network (95 nodes, 298 links). The main author is marked in red. (a) authors who collaborated on a single paper, (b) main co-author, (c) papers co-authored by the three central nodes.

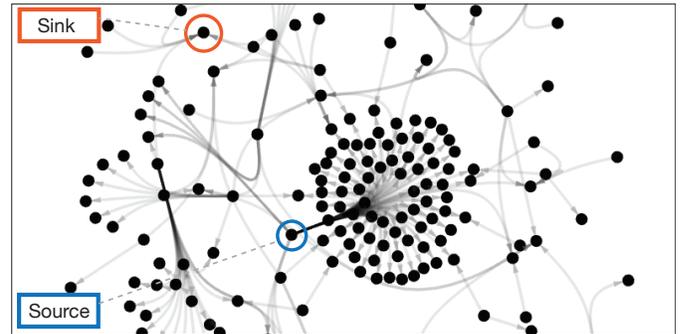


Fig. 7. Twitter network (456 nodes, 405 links). Detail of a central node who is retweeted a lot (source) and a node who retweets but is not retweeted by others (sinks).

4 EXPRESSION OF NETWORK MOTIFS AND ARTIFACTS

Having presented two examples of real-network renderings, we now investigate how to *read* CDs, in a more detailed and systematic way; we report on visual patterns generated by CDs; patterns that reveal information on the underlying graph topology, and artifacts that may imply wrong information. To structure this discussion, we propose to systematically review network motifs and present associated visual patterns in CD generated by our technique. When appropriate, we discuss how these motifs are expressed in spatial edge bundling.

4.1 Motifs

Motifs are informative topological structures in networks and can reveal insights on the organization and function of a phenomena represented by the network. Many motifs consist of very small structures such as dyads and triads, important for example in the social sciences [43]. Analyzing systematically the variation of sub-structures rapidly increases as the directionality of links adds to the complexity (e.g. 13 isomorphic triadic structures in a 3-node subgraph). Researchers across different fields identified a set of larger motifs [3], revealing meaningful structures such

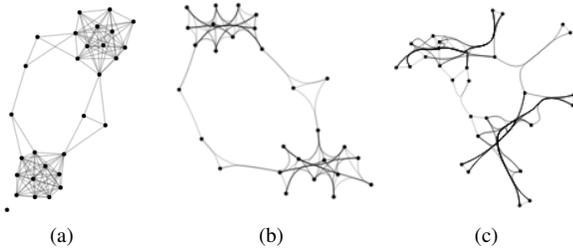
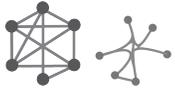


Fig. 8. Clusters in functional brain connectivity: (a) node-link diagram with force-directed layout, (b) node-link with edge bundling, (c) CD.

as regulatory mechanisms in biology [30]. This complexity makes the automatic extraction and systematic analysis of large motifs challenging. Network visualizations can provide a solution to help identify salient motifs in networks [10, 22].

Ideally, edge-bundling techniques such as CD can help reveal higher-level motifs in node-link diagrams when they are present in the network but difficult to identify due to edge crossings or clutter. Below, we provide a visual benchmark for CD and related techniques, discussing visual patterns for 9 most common motifs, collected from biology, sociology and network science literature. A more complete data set of comparing edge-bundling techniques is available on our website.



CLUSTERS are subgraphs composed of nodes that are densely connected to each other. There is no strict definition of cluster (e.g. no density threshold).

This motif is perhaps one of the most studied and many algorithms exist in different fields to identify them (e.g., [6, 14]). A challenge for identifying this motif is the relatively loose definition and the fact that in practice, cluster often overlap, with individual nodes belonging to two or more at the same time.

Visually, clusters in node-link diagrams with force-directed layouts usually resemble a dense area with the respective cluster nodes being spatially close and connected by many links crossing each other (Fig. 8(a)). By bundling edges within these clusters, edge bundling and CD visually “collapse” clusters into strings of bundles with sub-bundles branching in and out. This results in a visual pattern rather counter-intuitive to the classic notion of visually dense areas and may make it rather difficult to identify clusters (Fig. 8(b)(c)). More precisely, very dense clusters in CD collapse into a single fractal-like structure, while clusters with spatial bundling may yield lattice-like structures (Fig. 9).



CLIQUEs are a special case of clusters, in which all nodes are connected to each other, without any missing connection. Cliques have been particularly studied in social sciences to depict communities of people who collaborate or communicate fully with one another. A clique can also comprise the entire network, called a *fully-connected* network.

Visually, cliques or complete networks reveal a crucial artifact in bundling techniques (Fig. 9). More than for clusters, both visual edge-bundling and CD generate visual pattern that appear to show topological structure where there is no such in the network. In the case of CD, the power-graph decomposition induces a random grouping of pairs of nodes, and recursive bundling as the modules are computed. This particular image may suggest that the graph has an underlying structure composed of major backbones, which is not the case. While a reader can still follow connections between each pair of nodes, mitigating these artifacts is difficult, and it remains open if training viewers can totally overcome this issue. Slightly relaxing bundles (Sect. 3.3) can be one option to better highlight clusters.

For directed graphs, CD deliver a quite different pattern for cliques, showing two main bundles, one for each edge direction (Fig. 10(a)). Gradually removing links quickly destroys this clean structure, resulting in many visually overlapping bundles. Consequently, nicely bundled subgraphs can indicate complete connectivity.

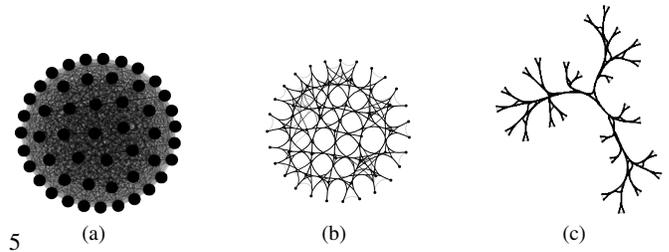


Fig. 9. Clique (complete network) in un-directed network: (a) node-link diagram with force-directed layout, (b) node-link with edge bundling, (c) CD.

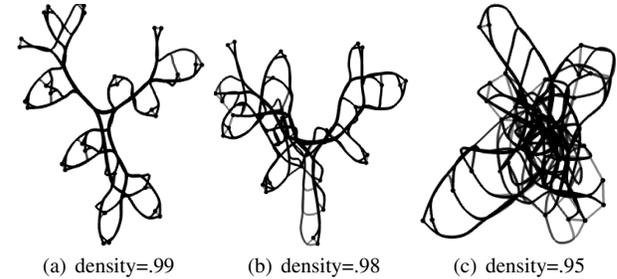


Fig. 10. Clique in directed network (a), and the decay of the pattern by gradually removing links (reducing edge density) (b,c).



BICLIQUES and **N-PARTITE COMPONENTS** are sets of nodes not connected within the same set, but connected across sets [31]. These motifs are particularly compelling when analyzing n-partite networks, which encode relationships between nodes of n different types (e.g. scientific authorship network composed of papers, authors, and keywords).

Visual patterns revealing bicliques are particularly salient in CD, compared to other techniques (Fig. 11). This directly results from the power-graph decomposition used to compute CD; sets of nodes with similar connections are grouped into modules and the links between modules are bundled together. Similar to the clique pattern; directed graphs which show two main bundles; and gradually removing edges rapidly destroys the typical “double-palm-tree” pattern (Fig. 12).

In force-directed node-link diagrams, n-partite components can be visually hidden due to missing connections between the individual nodes that move nodes of the same module (or set) closer together (Fig. 13). Though the double-palm pattern also appears through spatial edge bundling, it does not necessarily indicate bicliques.



STARS are subgraphs in which one node is connected to a set of mutually non-connected nodes (e.g. a professor co-authoring with his students).

Directed graphs further differentiate between *sinks* (mostly incoming connections) and *sources* (mostly outgoing connections).

Star motifs are highly visible in CD as power-graph decomposition groups all neighbors of a central node into a module, causing all links to be bundled together and rendered as a palm-tree pattern (Fig. 14(a)). In directed graphs, the neighbors are divided into two groups, and links bundled for each direction (incoming/outgoing edges).



TREES are subgraphs in which nodes are hierarchically organized (from root node to children nodes to leaves). Trees can be seen as chained star motifs, organizational hierarchies or networks of propagation such as messages on twitter. A more loose definition of trees include genealogical networks. Trees do generally not cause artifacts in edge-bundling techniques and, similar to stars, are relatively salient in CDs (Fig. 14(e)).

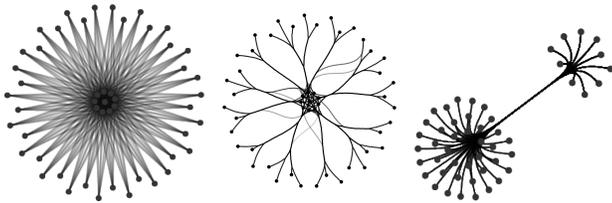


Fig. 11. Stars and bicliques: (a) node-link diagram with force-directed layout, (b) node-link with edge bundling, (c) CD.

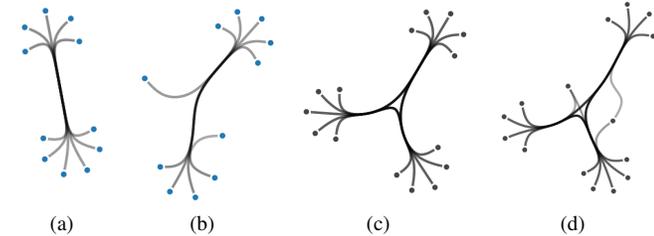
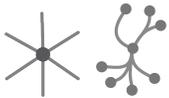


Fig. 12. N-clique network motifs degeneration (complete and one missing edge): (a,b) bi-clique, (c,d) 3-clique.



HUBS depict nodes with a high degree, compared to other nodes in the network. Hubs are highly studied in social sciences and are a main characteristic of scale-free networks in which the degree distribution follows a power-law distribution [7].

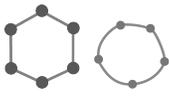
Because of their high number of connections, hubs in CD usually are connected to multiple bundles. Fig. 14 shows two different hubs: (a) part of its neighbors are not connected (similar to a star), and (b) neighbors are connected to many nodes, resulting in 4 bundles attached to the hub. The bundling effect makes it generally hard to estimate the degree of node, and thus, to identify hubs quickly. Viewers may learn to identify nodes with a high degree as connected to dark bundles when opacity encodes number of links, but hubs likely remain less salient than in standard node-link diagrams.

In directed graphs, similar to stars, bundles are divided into incoming and outgoing edges. Thus, a hub with a single bundle is either a source, or a sink, depending on the direction of the edges. Fig. 7 gives two examples of these types of pattern. CD is particularly compelling compare to standard node-link diagrams for identifying these types of patterns in directed graphs.



ARTICULATION POINTS are nodes that, if removed from the network, lead to two or more disconnected components. This motif is also indicative of the centrality of a node, and crucial when analyzing communication or transmission networks as information has to flow through this node to reach the individual components.

Since the layout of CD is based on the routing graph, articulation points are as visible as they are in straight node link diagrams (see nodes connecting clusters in Fig. 8). For those cases where articulation nodes are placed in the empty space between the clusters it connects, we cannot find any major difference between the edge-bundling techniques. Problems arise if articulation points are overlapping with other graph elements. In these cases, if a node is connecting two clusters, it still appears as node with strong bundles attached to it (strong bundles are darker in our renderings). Again, visual edge-bundling techniques will not make a difference between a point being connected to the surrounding nodes, and a point being connected to distant nodes.



PATHS and **CYCLES** are subgraphs in which nodes are sequentially connected to each other. If a path is closed, it is called a *cycle*. This motif is particularly important in communication or transmission networks, as it reveals how

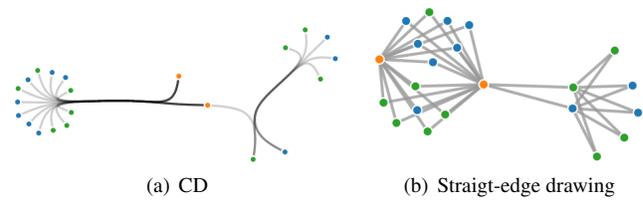


Fig. 13. Genealogy network (parent-child relations) with bi-cliques (green=male, blue=female, orange=unknown).

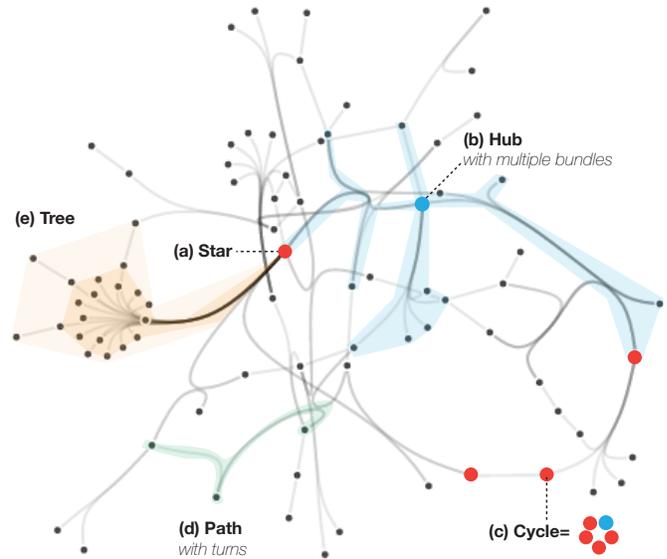


Fig. 14. This networks depicts migration between nations. We annotated visual patterns in CD corresponding to 5 motifs: (a) star, (b) hub, (c) cycle, (d) path, (e) tree.

information or diseases spread in the network. CDs makes certain paths and cycles visually salient as they appear as arcs with several edges and bundles, spanning large parts of a network (Fig. 14(c)). Curvature and bundling make it easy to follow paths as the eye of the viewer can follow curves according to the Gestalt principle of continuation [20]. However, some paths and cycles can contain sharp turns (Fig. 14(d)), making it harder to perceive the continuity of the path. Other more subtle perception mechanisms at play in CD are related to the saliency of some curves over others (for example in Fig. 15).

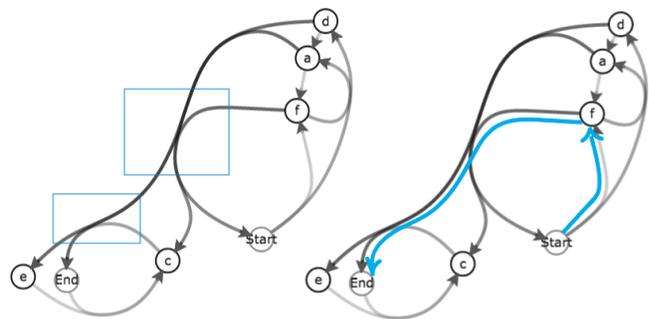


Fig. 15. Rectangles indicate regions causing the curve from *d* to *e* to be more visually salient. Thus, when looking for a path from *start* to *end*, the blue path may be less likely to be found than the path via *d*.

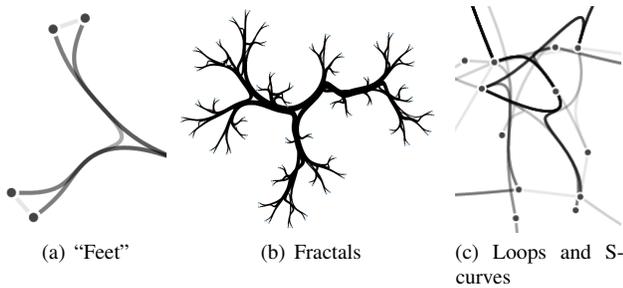


Fig. 16. Artifacts in Confluent drawings.

4.2 Artifacts

To summarize, some network motifs are visually salient in CD, for example, stars, paths, and n -cliques. However, these motifs become less evident if interweaved with too many other motifs. Other network motifs, including clusters and hubs, may be hard to perceive or require learning or even additional visual support.

Besides meaningful motifs, we encountered visual artifacts that wrongly imply topological structures in the network. Similar pattern can be also found in spatial edge bundling (e.g. [19, 21, 29, 41]). In CD, artifacts may be introduced by the power-graph decomposition or the way we create and lay out the routing graph. Fig. 16 summarizes common artifacts we observed.



FEET are patterns of two nodes sharing a bundle (Figure 16(a)) and being connected to each other. In the simplest case both nodes are connected to one common neighbor. In other cases they connected bundle is part of a larger bundle. In this case, feet artifacts are a result of the power-graph decomposition of densely connected components; feet artifacts describe two nodes in a density connected component.



FRACTALS have been discussed as resulting from cliques that collapsed into a single bundle with fractal-like sub-bundles (Figure 16(b)). Though visually, fractals imply a tree-topology in the network.



LOOPS and S-CURVES are bundles with strong changes in directionality, somewhat breaking the visual continuity of curves (Fig. 16(c)). Both artifacts occur in dense subgraphs that are too sparse to be collapsed into a single bundle (also see Fig. 10).

5 READABILITY STUDY

Last section gave an overview of how to read CDs on a motif level. Now, we are interested in the general readability and learnability of confluent graphs on a lower level: “Can users correctly perceive links in CDs?”. To that end, we report on a controlled user study that focuses on the following questions:

- Q1:** Can people with no expertise in graph theory *learn to read confluent drawing* to perform low-level readability tasks?
- Q2:** While we know traditional bundling techniques introduce ambiguities, can we *quantify how error-prone these techniques are* for low-level readability tasks?
- Q3:** The confluent drawing method we propose relies on the same power-graph decomposition as the state-of-the-art edge compression technique presented in [13]. However CD bundles edges between nodes of different groups, whereas edge compression visually exposes these groups, nesting them if necessary, and replaces edges between individual nodes by edges between modules. *Is there a difference in performance between both techniques?*
- Q4:** Finally, we are interested in users’ *subjective preferences* among power graphs and bundling techniques.

5.1 Techniques

We used a within-subject design, in which all participants performed the task with all techniques, counterbalancing their order of appearance. We compared four different techniques (implementations taken from the respective sources), all designed to reduce link clutter in node-link diagrams. Fig. 17 shows an example network from the study, rendered in all four techniques:

MB: Metro-style bundling is closest to the familiar straight-line drawings but removes some clutter. Described in [37], it uses two techniques to reduce clutter and ambiguities in flat node-link diagrams: (i) routing edges around nodes that are not connected to that edge, and (ii) bundling edges that follow similar trajectories but spacing the individual links so they remain distinct (e.g. similar to rendering of common lines on metromaps) (Fig. 17(a)).

PG: In the Power graph visualization [13], nodes with similar neighborhoods are grouped and their individual edges replaced by edges between groups (no information loss (Fig. 17(b))).

EB: We selected a representative implementation [29] for spatial edge bundling which bundles edges with similar trajectories for directed graphs. Note that this condition used a standard force-directed layout. To limit ambiguities so tasks were achievable, we relaxed the bundling slightly so individual links became visible. However, we could not totally limit the ambiguities caused by bundling in difficult networks (see Fig. 17(c)).

CD: Finally, we used the method we describe earlier to generate CD, but did not use any relaxation (Fig. 17(d)).

Note that to maintain the duration of the study to a reasonable time, we opted to exclude flat node-link diagrams (similar to [13]) and instead, opted to compare three edge-bundling techniques and the state-of-the-art edge compression technique.

5.2 Task

Since the techniques we compare attempt to deal with the edge clutter problem, we selected a path-following task on directed graphs, which is likely to be the most affected type of task since it requires participants to follow a set of successive links in the network. As in previous experiments [13], we selected a shortest-path task. We provided participants with diagrams featuring a “start” and an “end” node. We varied the shortest-path length between these nodes, from 1 to 4 links. We also included trials where there was no possible path between nodes.

5.3 Dataset and Difficulty

We used the graphs generated in [13], which present a modular structure with some level of noise. As explained in this previous study, synthetic graphs following this structure allow for a fine control over the level of difficulty of the task for edge compression techniques as they allow control over the number of modules and their level of nesting.

Graphs were generated for three levels of difficulty, characterized by their size (7, 10 and 15 nodes), density (20, 30 and 50 links for each size) and the number of modules and level of nesting they generate when computing their power-graph decomposition. In a nutshell, *easy* graphs are the smallest and contain 2 or 3 modules, possibly including a single level of nesting; *medium* graphs contain 3 to 5 modules including a single level of nesting; and *difficult* graphs contain 5 or 6 modules including two levels of nesting (details in [13]). While all graphs we selected are rather small in number of nodes, the difficult graphs exhibit a high level of connectivity complexity (Fig. 17(a)).

5.4 Participants and Setup

We recruited 15 participants through a local recruiting service and the local university. None of the participants had expertise in graph theory. The investigator instructed participants with a paper printout before each technique and instructed them to answer correctly as fast as they could. Participants then proceeded to complete trials using the experimental software. For each technique, users first had to completed 12 training trials successfully ensuring they understood task and technique correctly. Then, they completed the following 18 trials of the technique.

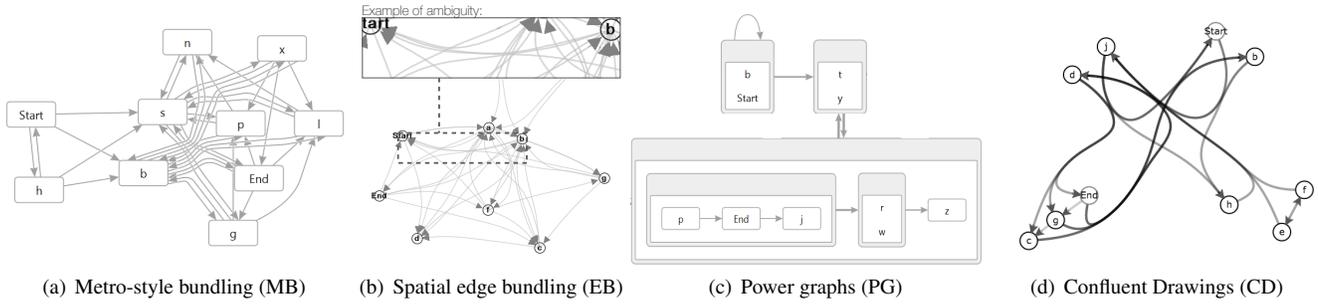


Fig. 17. Examples graph rendered in the four techniques compared in the path following task in our readability study.

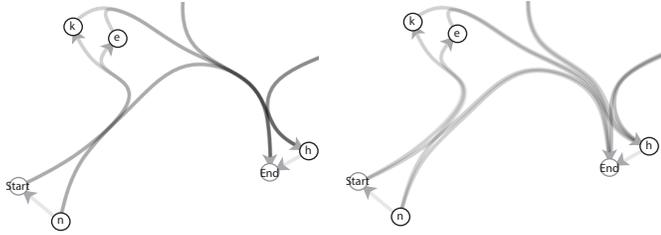


Fig. 18. Detail of network causing errors. Left: the direct link from start to end was overlooked and a path of 2 was instead reported. Right: relaxing bundles could be a solution.

Participants started a trial by pressing a button, which displayed the diagram and started the timer. Participants completed the task visually and pressed the space bar as soon as they found the answer. After pressing the space bar, the timer stopped and the diagram disappeared, preventing participants from checking their answers. Participants then answered a multiple choice question and could choose to have a break before starting the next trial. The investigator was observing from a different room to avoid distracting participants. After completing the four blocks, participants completed a preference questionnaire and gave feedback to the investigator.

5.5 Results

Accuracy: Since accuracy does not follow a normal distribution, we used Friedman’s non-parametric test. Friedman’s test revealed significant differences in accuracy across Technique ($p < .005$). Wilcoxon’s signed test on paired techniques revealed significant differences between the power-graph edge compression (PG) technique and both CD ($p < .002$) and EB ($p < .004$). PG, with a mean accuracy of 86% ($SE=2.1$) proves significantly more accurate than either CD or EB, both with a mean accuracy of 77% ($SE=2.5$) (Fig. 19(a)).

(Q1) Overall, we found that participants could learn CD to perform path-following tasks as well as in the other two edge-bundling techniques. Since other bundling techniques do not require participants to learn specific rules, this result is interesting. A single participant appears to have issues learning the technique and only achieved 50 percent correct trials with CD. However, contrary to our expectations, CD did not prove more accurate than edge bundling (EB). Looking at accuracy, split by the difficulty of the datasets, reveals that participants made more errors in difficult graphs (Fig. 19). We reviewed all errors made by the participants with CD and hypothesized that the perceptual phenomenon of some paths being more salient than others (described in Fig. 18) may be the cause of many errors; e.g., 6 participants failed to see the shortest path (a direct link) in Fig. 18 and instead reported a path of 2. To give an idea of the proportion of these errors, relaxing the task to identify reachability (e.g. is there a path of any length between start and end nodes) reduces the error rate from 23% to 8% in CD.

(Q2) Our results also reveal that spatial edge bundling (EB) is not more error-prone than the other bundling techniques for the graphs we tested. This finding may only apply to bundling techniques with *some* relaxation but show that bundling techniques can convey information with similar accuracy levels for small (but complex) graphs.

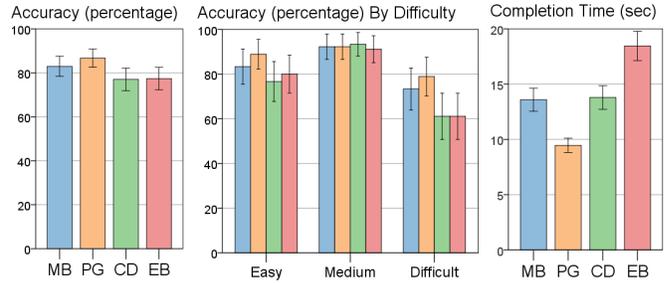


Fig. 19. Study results: mean accuracy per technique (left), mean accuracy by dataset difficulty (middle), mean completion time (right).

(Q3) The results also revealed that CD is more error-prone than PG. We relate this to the fact that power graphs (PG) contain generally less visual edges to be potentially followed and hence cause less visual clutter. Once power graphs are learned they can be very efficient.

Completion time: Since the distribution of completion time was skewed, we analyzed the logarithm of completion time as is common practice. We performed a repeated-measure analysis of variance (RM-ANOVA) for all trials as well as a mixed linear model (MLM) analysis for correct trials only (excluding 19% of the trials). Both analyses revealed the same significant differences between techniques.

We only report the result for correct trials. MLM reveals a significant difference in completion time across Technique ($F = 68.75, p < .0001$). Pairwise comparison indicates that PG significantly outperforms the other three techniques ($p < .0001$) and that EB significantly underperforms the other three techniques ($p < .0001$). Participants spend on average 9.3 seconds per trial ($SE = 0.3$) using PG, 12.8 seconds ($SE = 0.6$) using CD, 13.5 seconds ($SE = .5$) using MB and 17.2 seconds using EB (Fig. 19(b)).

(Q2) The completion time result confirms that edge bundling (EB) required most effort to conduct path-following tasks. We hypothesize that the longer completion time is due to participants attempting to resolve ambiguities (Fig. 17(b)-detail).

(Q3) The results confirmed that power graphs (PG) outperform all bundling techniques (MB, EB and CD). While the technique was not *significantly* more accurate than the metro-style bundling (MB), the reduced number of edges allowed participants to complete the task faster, demonstrating that power graph is the most effective technique for the task and graphs we tested.

Subjective User Feedback: After the user study, we asked users to rate each techniques on a 0 (very bad) - 4 (very good) Likert scale. We used the following questions:

- How would you rate the technique overall? (**Overall**)
- How easy was the technique to learn? (**Learnability**),
- How clutter-free was the technique? (**Clutter-free**),
- How confident did you feel with the technique? (**Confidence**),

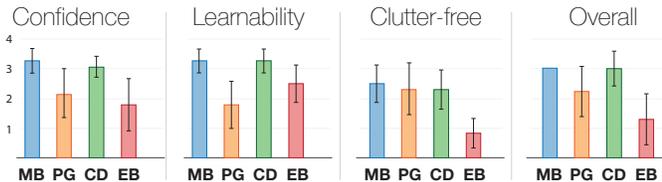


Fig. 20. Reported user preferences.

Fig. 20 shows graphs of participants’ ratings for our questions. Here, we report salient insights from participants’ answers. Participants agreed most on their ratings for MB and CD and diverged more for PG and EB. Overall, MB and CD were the most preferred techniques for path-following tasks, over PG and EB.

(Q4) Participants’ answers indicated that they consider MB and CD as the most learnable, and PG to be the least learnable technique. While we expected the lower ranking for the PG technique based on the result of [13], we were surprised that CD was ranked similarly to MB. Indeed, MB is the closest to flat node-link diagrams that participants have certainly encountered before and does not require specific rules to learn in order to follow edges. Therefore, we would have expected CD to rank lower. Participants’ answers on clutter were also surprising as they ranked PG, MB and CD techniques similarly. While we expected EB to appear the most cluttered (especially regarding the relaxation), we did not expect MB to be ranked as high.

Perhaps the most interesting result is about confidence in the tasks completed with each technique. Despite the high accuracy of PG, participants varied greatly in their confidence in the technique. Many participants rated their confidence with PG lower than with CD, despite their actual accuracy being significantly lower with CD than with PG. This may be explained by ratings in learnability, since PG was generally found the most difficult to understand.

6 DISCUSSION AND FUTURE WORK

The results of our study indicate that CD is an understandable technique for novices (Q1) and it has some advantages over the regular edge bundling technique: it allows for similar accuracy but faster completion time and is generally preferred by participants. However, our study did not reveal significant differences between CD and metro-style bundling.

The analysis of errors participants made in CD pointed to artifacts in perception of some paths over others. This effect may be emphasized by the study instructions asking participants to answer as fast as they could. Thus, it is possible that they answered as soon as they found a path between the nodes (and did not check if there was a shorter one). However, participants demonstrated a high confidence in their answers, which indicates that CDs can prove misleading.

The results also confirmed that EB was less effective (Q2) in terms of task completion. The technique did not cause more errors than other bundling techniques overall but still required participants a longer time to find paths in the network. Note that despite the slight bundling relaxation that we used to limit connectivity ambiguities, several ambiguities remained in densely connected areas of the networks (Fig. 1(b)), probably causing participants additional time to resolve them.

To conclude our study: the most effective technique for path-following tasks are power-graphs (PG) as they outperformed all other tested bundling techniques, including CD, a technique based on the same underlying power-graph decomposition (Q3). However, it is important to note that power graphs require more learning than other techniques, and participants have generally lower confidence in their interpretations (Q4). Presumably, they were the most unfamiliar looking rendering and using grouping to imply structural equivalence cannot be said to be particularly favored by participants.

6.1 Future Studies

As with all controlled studies, our results should not be generalized beyond the techniques (including alternative spatial edge-bundling techniques), tasks, and graph characteristics we tested. While the graphs tested were small (up to 15 nodes), the level of complexity of their connectivity is what we wanted to explore in this study. Future

studies should investigate larger graphs and especially focus on higher-level readability tasks such as, for example, identifying motifs as we described in Section 4. We also envision further user studies on the readability of other common edge bundling and edge compression techniques, but also adjacency matrices, and specific encodings of edge direction such as detailed in [25].

In general, higher-level exploration tasks for networks, such as comparison of networks, exploring evolving networks over time, or assessing density and counting number of clusters, are underexplored. Understanding how edge compression and bundling artifacts affect these tasks needs further study and the techniques themselves may need adaptation. Our survey of motifs is a first attempt into a direction which provokes further questions: which motifs are represented by which visual patterns?; which motifs are preserved and most salient?; and which graph characteristics (e.g., density, scale-freeness, path-length) are preserved best with which network visualization technique?

6.2 Improving Confluent Drawings

While our visual benchmark was entirely based on the shape and saliency of motifs, there may be other ways to compare bundling techniques prior to user studies or to inform drawing heuristics. For example, one could try measuring the number of bundles; quantifying angles, length, and number of edges in bundles, or measure the number of edge crossings after bundling. Drawing heuristics could be used to improve the respective drawings and bundlings. For example, we can imagine splitting specific bundles that constrain the layout; either interactively or based on heuristics measuring the “stress” on a bundle.

While our algorithm for drawing CDs is a first attempt and meant to render readable drawings, we imagine extending the possible drawings by trying to create bundling based on more complete (non-hierarchical) bipartite clique detection and to introduce specific renderings for specific motifs. Though we experimented with the rotary-glyph for cliques proposed in [9], clusters were decomposed into cliques, which lead to too many overlapping clique glyphs.

Furthermore, spatial edge bundling works on a *given* layout, while power graphs and confluent drawings both create their *own* layout. One can imagine drawing confluence drawings on top of force-directed layouts of the graphs themselves (i.e. not the routing graph) or hybrid techniques; for example, using power graph-drawings for dense areas, while bundling and edges between power graphs using CD. CDs could also be augmented with edge bundling and edge routing to avoid nodes being incidentally placed on bundles.

We are especially intrigued by the particle animations simulating flow along edges and bundles. We believe this technique opens the field for future research in visualizing flow and propagation in networks representing transport, diseases, or messages. We believe CDs provide a promising basis from which to start, since animations along curves appear much less cluttered than with regular straight-line drawings of node-link diagrams.

7 CONCLUSION

We conclude that confluent drawings present a readable format to reduce some edge clutter in node-link representations. The fact that confluent drawings are based on network topology makes them an interesting alternative to spatial edge bundling, though new ambiguities may occur and that require more studies and research. In providing a first working implementation to render and explore confluent drawings, we were able to observe patterns and artifacts and to give an impression of the potentials and drawbacks of CDs for drawing real-world networks.

We found that CDs work best for generally sparse networks, networks with locally dense clusters (certain social networks), and those networks with a certain degree of structure in the form of bicliques (e.g. multi-modal networks), cycles, stars, or trees patterns (e.g. genealogy). CDs do result in cluttered images for significantly dense (but not complete (Fig. 10)) network that lack the just mentioned structures. We found CDs working well for both directed and undirected networks. For large graphs, visual clutter highly depends on the graph structure; bigraphs and cliques get collapsed into bundles (e.g., Fig. 10(a)), while other structures will cause significant clutter (e.g., Fig. 10(c)).

REFERENCES

- [1] cola.js home page: <http://marvl.infotech.monash.edu/webcola/>.
- [2] J. Abello, F. Van Ham, and N. Krishnan. Ask-graphview: A large scale graph visualization system. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):669–676, 2006.
- [3] U. Alon. Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8:450–461, 2007.
- [4] D. Archambault, T. Munzner, and D. Auber. Grouseflocks: Steerable exploration of graph hierarchy space. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):900–913, 2008.
- [5] D. Archambault, H. C. Purchase, and B. Pinaud. The readability of path-preserving clusterings of graphs. In *Computer Graphics Forum*, volume 29, pages 1173–1182. Wiley Online Library, 2010.
- [6] E. Arias-Castro, G. R. Grimmett, et al. Cluster detection in networks using percolation. *Bernoulli*, 19(2):676–719, 2013.
- [7] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.
- [8] M. S. T. Carpendale and X. Rong. Examining edge congestion. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '01, pages 115–116, New York, NY, USA, 2001. ACM.
- [9] M. Dickerson, D. Eppstein, M. T. Goodrich, and J. Y. Meng. Confluent drawings: Visualizing non-planar diagrams in a planar way. *J. Graph Algorithms Appl.*, 9(1):31–52, 2005.
- [10] K. Dinkla, M. A. Westenberg, and J. J. van Wijk. Compressed adjacency matrices: Untangling gene regulatory networks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2457–2466, Dec 2012.
- [11] C. Dunne and B. Shneiderman. Motif simplification: Improving network visualization readability with fan, connector, and clique glyphs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 3247–3256, New York, NY, USA, 2013. ACM.
- [12] T. Dwyer, C. Mears, K. Morgan, T. Niven, K. Marriott, and M. Wallace. Improved optimal and approximate power graph compression for clearer visualisation of dense graphs. In *Visualization Symposium (PacificVis), 2014 IEEE Pacific*, pages 105–112. IEEE, 2014.
- [13] T. Dwyer, N. H. Riche, K. Marriott, and C. Mears. Edge compression techniques for visualization of dense directed graphs. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2596–2605, 2013.
- [14] T. Eltoft and R. J. DeFigueiredo. A new neural network for cluster-detection-and-labeling. *Neural Networks, IEEE Transactions on*, 9(5):1021–1035, 1998.
- [15] D. Eppstein, M. T. Goodrich, and J. Y. Meng. Delta-confluent drawings. In *Graph Drawing*, pages 165–176. Springer, 2005.
- [16] D. Eppstein, M. T. Goodrich, and J. Y. Meng. Confluent layered drawings. *Algorithmica*, 47(4):439–452, 2007.
- [17] D. Eppstein, D. Holten, M. Löffler, M. Nöllenburg, B. Speckmann, and K. Verbeek. Strict confluent drawing. In *Graph Drawing*, pages 352–363. Springer, 2013.
- [18] D. Eppstein and J. A. Simons. Confluent hasse diagrams. In *Graph Drawing*, pages 2–13. Springer, 2011.
- [19] O. Ersoy, C. Hurter, F. Paulovich, G. Cantareiro, and A. Telea. Skeleton-based edge bundling for graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2364–2373, Dec 2011.
- [20] D. Field, A. Hayes, and R. Hess. Contour integration by the human visual system: evidence for a local association field. *Vision Research*, 1993.
- [21] E. R. Gansner, Y. Hu, S. North, and C. Scheidegger. Multilevel agglomerative edge bundling for visualizing large graphs. In *2011 IEEE Pacific Visualization Symposium*, pages 187–194, March 2011.
- [22] N. Henry and J.-D. Fekete. MatrixExplorer: a Dual-Representation System to Explore Social Networks. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):677–684, Sept 2006.
- [23] M. Hirsch, H. Meijer, and D. Rappaport. Biclique edge cover graphs and confluent drawings. In *Graph Drawing*, pages 405–416. Springer, 2006.
- [24] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, Sept. 2006.
- [25] D. Holten, P. Isenberg, J. J. van Wijk, and J. D. Fekete. An extended evaluation of the readability of tapered, animated, and textured directed-edge representations in node-link graphs. In *Visualization Symposium (PacificVis), 2011 IEEE Pacific*, pages 195–202, March 2011.
- [26] D. Holten and J. J. van Wijk. Force-directed edge bundling for graph visualization. In *Proceedings of the 11th Eurographics / IEEE - VGTC Conference on Visualization*, EuroVis'09, pages 983–998, Aire-la-Ville, Switzerland, Switzerland, 2009. Eurographics Association.
- [27] P. Hui, M. Schaefer, and D. Štefankovič. Train tracks and confluent drawings. In *Graph Drawing*, pages 318–328. Springer, 2004.
- [28] C. Hurter, O. Ersoy, S. I. Fabrikant, T. R. Klein, and A. C. Telea. Bundled visualization of dynamicgraph and trail data. *IEEE Transactions on Visualization and Computer Graphics*, 20(8):1141–1157, 2014.
- [29] C. Hurter, O. Ersoy, and A. Telea. Graph Bundling by Kernel Density Estimation. *Computer Graphics Forum*, (31):865–874, 2012.
- [30] T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J.-B. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford, and R. A. Young. Transcriptional regulatory networks in *saccharomyces cerevisiae*. *Science*, 298(5594):799–804, 2002.
- [31] S. Lehmann, M. Schwartz, and L. K. Hansen. Biclique communities. *Phys. Rev. E*, 78:016108, Jul 2008.
- [32] S.-J. Luo, C.-L. Liu, B.-Y. Chen, and K.-L. Ma. Ambiguity-free edge-bundling for interactive graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 18(5):810–821, 2012.
- [33] M. E. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.
- [34] Q. Nguyen, P. Eades, and S.-H. Hong. On the faithfulness of graph visualizations. In *Visualization Symposium (PacificVis), 2013 IEEE Pacific*, pages 209–216. IEEE, 2013.
- [35] V. Peysakhovich, C. Hurter, and A. Telea. Attribute-driven edge bundling for general graphs with applications in trail analysis. *Visualization Symposium (PacificVis), 2015 IEEE Pacific*, pages 39–46, 2015.
- [36] D. Phan, L. Xiao, R. Yeh, P. Hanrahan, and T. Winograd. Flow map layout. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, INFOVIS '05, pages 29–, Washington, DC, USA, 2005. IEEE Computer Society.
- [37] S. Pupyrev, L. Nachmanson, S. Bereg, and A. E. Holroyd. Edge routing with ordered bundles. In *Proc. 19th Internat. Sympos. on Graph Drawing*, LNCS 7034, pages 136–147. Springer-Verlag, 2011.
- [38] G. Quercini and M. Ancona. Confluent drawing algorithms using rectangular dualization. In *Graph Drawing*, pages 341–352. Springer, 2010.
- [39] L. Royer, M. Reimann, B. Andreopoulos, and M. Schroeder. Unraveling protein networks with power graph analysis. *PLoS Comput Biol*, 4(7):1–17, 07 2008.
- [40] D. Selassie, B. Heller, and J. Heer. Divided edge bundling for directional network data. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2354–2363, 2011.
- [41] M. van der Zwan, V. Codreanu, and A. Telea. Cubu: Universal real-time bundling for large graphs. *IEEE Transactions on Visualization and Computer Graphics*, PP(99):1–1, 2016.
- [42] C. Ware, H. Purchase, L. Colpoys, and M. McGill. Cognitive measurements of graph aesthetics. *Information Visualization*, 1(2):103–110, June 2002.
- [43] S. Wasserman and K. Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.
- [44] N. Wong, S. Carpendale, and S. Greenberg. Edgelens: An interactive method for managing edge congestion in graphs. In *Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on*, pages 51–58. IEEE, 2003.
- [45] H. Zhou, P. Xu, X. Yuan, and H. Qu. Edge bundling in information visualization. *Tsinghua Science and Technology*, 18(2):145–156, April 2013.